

19990723 033



IMPLEMENTATION OF A TWO-DIMENSIONAL
HYDRODYNAMIC SHOCK CODE BASED UPON
THE WEIGHTED AVERAGE FLUX METHOD

THESIS

Mark P. Wittig, CPT, USA
AFIT/GAP/ENP/99J-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Approved for Public Release; distribution unlimited

AFIT/GAP/ENP/99J-02

**IMPLEMENTATION OF A TWO-DIMENSIONAL
HYDRODYNAMIC SHOCK CODE BASED UPON
THE WEIGHTED AVERAGE FLUX METHOD**

THESIS

MARK P. WITTIG, CAPTAIN, USA

AFIT/GAP/ENP/99J-02

Approved for Public Release; distribution unlimited

**IMPLEMENTATION OF A TWO-DIMENSIONAL
HYDRODYNAMIC SHOCK CODE BASED UPON
THE WEIGHTED AVERAGE FLUX METHOD**

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Mark P. Wittig, B.S.

Captain, USA

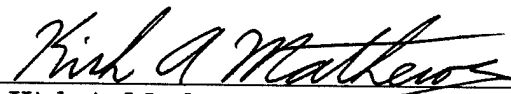
June 1999

Approved for Public Release; distribution unlimited

IMPLEMENTATION OF A TWO-DIMENSIONAL HYDRODYNAMIC SHOCK CODE BASED UPON THE WEIGHTED AVERAGE FLUX METHOD

Mark P. Wittig, B.S.
Captain, USA

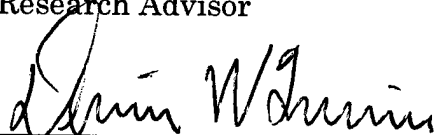
Approved:



Kirk A. Mathews
Associate Professor of Nuclear Engineering
Research Advisor

8 June 1999

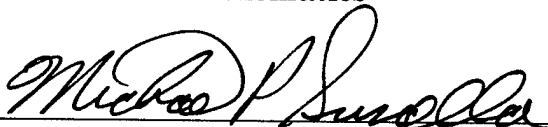
Date



Dennis W. Quinn
Professor of Mathematics

7 JUNE 99

Date



Michael P. Susalla
CDR, USN
Instructor of Nuclear Engineering

7 June 99

Date

Acknowledgements

There are many people that contributed to success of this research effort. First and foremost, I thank my wife, Ivonne, daughter, Jennifer, and son Mathew, for their unwavering support, understanding and patience over the last two years at AFIT. Second, I thank Dr. Kirk Mathews for his guidance and support. From whom I have truly learned a great deal. Additionally, I thank my committee, Dr. Dennis Quinn and CDR Michael Susalla for their time and support. Lastly, I thank my peers who listened and offered sound advice throughout this work.

Table of Contents

Acknowledgements	iii
List of Figures	viii
List of Tables	xi
List of Symbols	xii
Abstract	xvii
Chapter I: Introduction.....	1
A. Motivation	1
B. Background	1
C. Problem	2
D. Scope	3
E. Assumptions and Limitations	3
F. Approach	3
Chapter 2: Governing Equations and Important Thermodynamic Relations ..	5
A. Simplifying Assumptions	5
B. Governing Equations	7
C. Important Thermodynamic Relations	10
Chapter 3: The Riemann Problem and Its Solution	13
A. The Riemann Problem	13
B. General Wave Structure of the Riemann Problem.....	14
C. Important Wave Relations	19
D. Toro's Exact Riemann Solver (ERS)	21

The Star Step	22
The Sample Step	24
The Rarefaction Step	26
E. The Sod Shock Tube Problem	28
Chapter 4: One-dimensional Shock Code Implementation	33
A. Initial Boundary Value Problem	33
B. Discretization of the Domain	34
C. Conservative Discretization of the Euler Equations	35
D. Godunov's Method	37
E. Implementation of Godunov's Method	39
Step 1: Phantom Cells and Boundary Conditions	41
Step 2: Determination of a Stable Time Step	43
Step 4: Sonic Rarefactions – A Special Case	45
Step 6: Update Cell Solution	46
F. Illustration of Godunov's Method	46
G. Improvements in Computational Efficiency and Accuracy	48
Adaptive Riemann Solver (ARS)	48
Primitive Variable Riemann Solver (PVRs)	50
Two-Rarefaction Riemann Solver (TRRS)	51
Two-Shock Riemann Solver (TSRS)	52
Adaptive Riemann Solver Results	53
Adaptive Courant Coefficient	55

Adaptive Courant Results	57
H. Toro's Weighted Average Flux (WAF) Method	59
I. Illustration of WAF	62
J. Limited-WAF (LWAF)	63
K. Illustration of Limited WAF	68
L. Testing and Validation of the One-Dimensional Shock Code	71
Testing	71
Validation	72
Chapter 5: Shock Code Implementation in Two Dimensions	75
A. Initial Boundary Value Problem	75
B. Discretization of Domain with Two Spatial Dimensions	75
C. Dimensional Splitting	77
D. Special Considerations	82
E. Sod Test in Two-Dimensions	85
Chapter 6: High Pressure Cylinder Problem	88
A. Problem	88
B. Shock Wave Propagation and Reflection over an Ideal Surface	89
Shock Wave Propagation	89
Shock Wave Reflection	91
C. Numerical Solution	94
Chapter 7: Conclusion	117
Appendix A: Toro's Initial Guess Generator	119

Bibliography	124
VITA	127

List of Figures

Figure 1: General Wave Structure of the Riemann Problem	16
Figure 2: Characteristics Defining a Left Traveling Rarefaction	16
Figure 3: Characteristics Defining a Right Traveling Shock	17
Figure 4: Characteristics Defining a Right Traveling Contact Surface	18
Figure 5: Exact Riemann Solver Solution Procedure	22
Figure 6: Riemann Problem Local Reference Frame	24
Figure 7: Typical Wave Structure	27
Figure 8: Density Profile at $t=0.25s$	30
Figure 9: Particle Velocity Profile at $t=0.25s$	31
Figure 10: Pressure Profile at $t=0.25s$	31
Figure 11: Specific Internal Energy Profile at $t=0.25s$	32
Figure 12: Discretized Domain	34
Figure 13: Arbitrary Mesh Rectangle	36
Figure 14: m^{th} Component of $\bar{U}(x, t^n)$ Mapped onto Discretized Spatial Domain	38
Figure 15: Godunov Solution Procedure	40
Figure 16: Phantom Cells	42
Figure 17: Wave Propagation Through Adjacent RPs	44
Figure 18: Godunov vs ERS - Density Profile at $t=0.25s$	47
Figure 19: Comparison of Godunov's Method Using ERS and ARS	54

Figure 20: Weights of WAF	60
Figure 21: WAF - Density Profile at $t=0.25s$	62
Figure 22: Calculation of Local Flow Parameters	66
Figure 23: LWAF - Density Profile at $t=0.25s$	69
Figure 24: Comparison of Limiters near Contact Surface	70
Figure 25: Comparison of Limiters near Shock	71
Figure 26: ARL Overpressure Time History at 31.44 m	73
Figure 27: Overpressure Time History at 31.44m	74
Figure 28: Pseudocode for 2D Code	80
Figure 29: WAF Wave Structure in Two-Dimensions	83
Figure 30: Limited WAF Wave Structure in Two-Dimensions	84
Figure 31: 2D Sod Density Profile at $t=0.25s$	86
Figure 32: Typical Overpressure Profile	91
Figure 33: Typical Shock Reflection Phenomena	92
Figure 34: Regular Reflection	93
Figure 35: Mach Reflection	94
Figure 36: Maximum Overpressure vs Time [0ms, 60ms]	97
Figure 37: Maximum Overpressure vs Time [9.84ms, 60ms]	99
Figure 38: Maximum Overpressure vs Ground Distance	100
Figure 39: High-pressure Cylinder Overpressure at $t=4.8ms$	103
Figure 40: High-pressure Cylinder Overpressure at $t=10.0ms$	104

Figure 41: High-pressure Cylinder Overpressure at $t=10.4ms$	105
Figure 42: High-pressure Cylinder Overpressure at $t=10.8ms$	106
Figure 43: High-pressure Cylinder Overpressure at $t=11.2ms$	107
Figure 44: High-pressure Cylinder Overpressure at $t=11.6ms$	108
Figure 45: High-pressure Cylinder Overpressure at $t=12.0ms$	109
Figure 46: High-pressure Cylinder Overpressure at $t=12.4ms$	110
Figure 47: High-pressure Cylinder Overpressure at $t=12.8ms$	111
Figure 48: High-pressure Cylinder Overpressure at $t=13.2ms$	112
Figure 49: High-pressure Cylinder Overpressure at $t=13.6ms$	113
Figure 50: High-pressure Cylinder Overpressure at $t=27.0ms$	114
Figure 51: High-pressure Cylinder Overpressure at $t=31.2ms$	115
Figure 52: High-pressure Cylinder Overpressure at $t=36.0ms$	116
Figure 53: General Behavior of Toro's Pressure Function	120

List of Tables

Table 1: Possible Wave Configurations.....	19
Table 2: Comparison Between the ERS and ARS.....	55
Table 3: Adaptive Courant Results - Sod Test.....	57
Table 4: Adaptive Courant Results - Sod Test.....	59
Table 5: High-pressure Cylinder Boundary Conditions	95
Table 6: High-pressure Cylinder Execution Parameters	96

List of Symbols

Symbol	Units	Description
t	s	Time
Δt	s	Time step
x	m	Spatial coordinate
Δx	m	Spatial step (x – direction)
y	m	Spatial coordinate
Δy	m	Spatial step (y – direction)
z	m	Spatial coordinate
L	m	Domain length one dimension
L_x	m	x – direction domain length
L_y	m	y – direction domain length
\hat{x}	m	Spatial coordinate in the Riemann Problem reference frame
n	–	Temporal index
N	–	Last temporal index
$X^{\Delta t}$	–	x – flow time step operator
$Y^{\Delta t}$	–	y – flow time step operator
i	–	Spatial index (x – direction)
I	–	Total number of cells in the x – direction
j	–	Spatial index (y – direction)
J	–	Total number of cells in the y – direction
u	m/s	Particle velocity (x – direction)
v	m/s	Particle velocity (y – direction)
w	m/s	Particle velocity (z – direction)
V_N	m/s	Particle velocity (<i>normal component</i>)
V_T	m/s	Particle velocity (<i>tangential component</i>)
c	m/s	Local speed of sound

Symbol	Units	Description
c_k	m/s	Local speed of sound in the k^{th} region
u_*	m/s	Particle velocity (x – <i>direction</i>) in the star region
u_k	m/s	Particle velocity (x -direction) in the k^{th} region
ζ	m/s	Similarity variable
λ_k	m/s	Speed of the k^{th} characteristic wave
S_k	m/s	Speed of the k^{th} wave
\tilde{S}_{\max}^n	m/s	Estimated maximum wave speed at t^n
S_{\max}^n	m/s	Maximum wave speed at t^n
$(\tilde{S}_i^n)_{\max}$	m/s	Estimated maximum wave speed at t^n in the x – <i>direction</i>
$(\tilde{S}_j^n)_{\max}$	m/s	Estimated maximum wave speed at t^n in the y – <i>direction</i>
\hat{S}	m/s	Reference speed at the point (\hat{x}, T)
ρ	kg/m ³	Density
ρ_k	kg/m ³	Density in the k^{th} region
E	J/m ³	Total energy density
e	J/kg	Specific internal energy
P_N	(kg-m/s)/m ³	Momentum (<i>normal component</i>)
P_T	(kg-m/s)/m ³	Momentum (<i>tangential component</i>)
p	Pa	Pressure
p_k	Pa	Pressure in the k^{th} region
p_*	Pa	Pressure star region
$p_*^{(k)}$	Pa	k^{th} iteration of the p_* root solver
p_{\min}	Pa	Minimum pressure in adjacent cells
p_{\max}	Pa	Maximum pressure in adjacent cells
Δp	Pa	Overpressure

Symbol	Units	Description
Δp^{\max}	Pa	Peak overpressure
Δp_R^{\perp}	Pa	Peak overpressure at the point of normal reflection
P_{pvrs}	Pa	Pressure as calculated using PVRs
s	J/kg-K	Specific entropy
R	J/kg-K	Specific gas constant
c_p	J/kg-K	Specific heat capacity at constant pressure
c_v	J/kg-K	Specific heat capacity at constant volume
γ	—	Ratio of specific heats (c_p / c_v)
T	K	Temperature
θ_I	degrees	Angle of incidence
θ_I^{crit}	degrees	Critical angle of incidence
\vec{U}	—	Vector of conserved variables
\vec{U}_i^n	—	Cell averaged solution in cell i and time t^n
$(\vec{U}^n)'$	—	Interim solution at t^n in cell (i, j)
\vec{F}	—	Flux of conserved variables in x - direction
$\vec{F}_{i+1/2}$	—	Flux through cell interface $i + 1/2$
$\vec{F}_{i-1/2}$	—	Flux through cell interface $i - 1/2$
\vec{F}^{GF}	—	Godunov numerical flux
\vec{F}^{WAF}	—	WAF numerical flux
\vec{F}^{LWAF}	—	LWAF numerical flux
\vec{G}	—	Flux of conserved variables in y - direction
\vec{H}	—	Flux of conserved variables in z - direction
\vec{W}	—	Vector of primitive variables
\vec{W}_k	—	Vector of primitive variables in the k^{th} region

Symbol	Units	Description
\bar{W}^{WAS}	—	The weighted average primitive variable state.
\bar{W}^{LWAS}	—	The limited weighted average primitive variable state.
C^-	—	Characteristic field with speed = $u - c$
C^0	—	Characteristic field with speed = u
C^+	—	Characteristic field with speed = $u + c$
$f_1(p^*, \bar{W}_1)$	—	Pressure function in region 1
$f_4(p^*, \bar{W}_4)$	—	Pressure function in region 4
$f_\xi(p^*, \bar{W}_\xi)$	—	Generalized pressure function
$d_{\xi \pm 1}(p^*, \rho_\xi, p_\xi)$	—	Generalized density function
R_*	—	Star region
R_k	—	k^{th} region in the Riemann Problem
C	—	Courant coefficient
C_{\max}	—	Maximum allowable Courant number
v_{\max}^n	—	Maximum local Courant number at t^n
v_k	—	Local Courant number of the k^{th} wave
$TV(\bar{U}^{n+1})$	—	Total variation function
$\phi_k(r, v)$	—	Weight limiter function of the k^{th} wave
$\psi_k(r)$	—	Flux limiter function of the k^{th} wave
σ_k	—	Sign function the k^{th} local Courant number
r_k	—	Local flow parameter of the k^{th} wave
$r_{i+1/2, k}$	—	Local flow parameter of the k^{th} wave at face $i + 1/2$
α_k	—	Wave amplifier of the k^{th} wave
w_k	—	Weight in the k^{th} region

Symbol	Units	Description
Q_s	–	Shock quotient
ε	–	User-defined error tolerance
I	–	Incident wave
R	–	Reflected wave
M	–	Mach Stem
T	–	Triple point

Abstract

Numerical modeling of shock propagation and reflection is of interest to the Department of Defense (DoD). Proprietary state-of-the-art codes based upon E. F. Toro's weighted average flux (WAF) method are being used to investigate complex shock reflection phenomena. Here we develop, test, and validate a one-dimensional hydrodynamic shock code. We apply WAF to Gudonov's first-order upwind method to achieve second-order accuracy. Oscillations, typical of second-order methods, are then removed using adaptive weight limiter functions based upon total variation diminishing (TVD) flux limiters. An adaptive Riemann solver routine is also implemented to improve computational efficiency. This one-dimensional code is then extended into two dimensions via Warming and Beam's variation on dimensional splitting. The numerical capabilities of the two-dimensional code are demonstrated by modeling the detonation of a cylindrically symmetric explosive with the axis of the cylinder oriented horizontally above an ideal surface.

Chapter I: Introduction

A. Motivation

Numerical modeling of shock propagation in hydrodynamic flows is a field of obvious interest to the Department of Defense (DoD). Some aspects are particularly difficult to model, for example, the transition from regular to Mach reflection. Recent state-of-the-art codes based upon E. F. Toro's weighted average flux (WAF) method are being used to investigate such shock reflection phenomena. Unfortunately, these codes are company-proprietary and unavailable for use within DoD. The objective of this effort is to develop our own WAF code for use at the Air Force Institute of Technology and throughout DoD. Ideally, we will find improvements in accuracy and/or computational efficiency in comparison with the algorithms used in other codes.

B. Background

In 1959, Godunov introduced an extension of the first-order upwind Courant-Isaacson-Rees scheme for solving the Euler Equations (LeVeque, 1992). This scheme was the first conservative upwind technique developed and has given rise to many of the modern numerical techniques in use today. Godunov's method assumes that a piecewise constant distribution of the conserved variables across the computational domain may be treated as a series of local Riemann problems. The flux is calculated by solving the Riemann problem exactly, forward in time. While robust, this technique is

computationally inefficient because each Riemann problem requires iteration to solve nonlinear equations. In an effort to make this scheme more efficient, several approximate Riemann solvers have been developed that solve the Riemann problem without requiring iterative processes. Modern Godunov-type methods are constructed in this fashion.

First-order schemes are known to result in solutions smeared at discontinuities due to numerical viscosity and Godunov's method is no exception. In 1989, E. F. Toro presented a new high-resolution technique that fully utilizes the wave structure of the Riemann problem to calculate a more accurate intercell flux. This flux is a weighted average of the flux vectors across the Riemann problem. Second-order accuracy is achieved in this way. Spurious oscillations typical of second-order schemes are removed using total variation diminishing (TVD) weight limiters (Toro, 1992).

Extension into two dimensions is easily achieved via dimensional splitting (Warming and Beam, 1976). In this approach, the two-dimensional initial value problem (IVP) solution is obtained by solving a sequence of one-dimensional problems in each coordinate direction. This technique is known as time-operator splitting.

C. Problem

The primary objective of this research is to develop a two-dimensional, computationally-efficient, hydrodynamic shock code based upon the weighted

average flux (WAF) method (Toro,1989) for the investigation of air blast phenomenology.

D. Scope

The scope of this project is limited to the modeling of air blast under ideal conditions.

E. Assumptions and Limitations

The computational model presented here assumes that air behaves as an ideal gas with a constant ratio of specific heats, i.e. $\gamma = 1.4$. This assumption limits use of the code to low pressure regimes where peak pressures are approximately 690 kPa or below.

F. Approach

Development of the two-dimensional code begins first with the introduction and application of all numerical techniques, except dimensional splitting, in one dimension. Code development is performed in a logical step-by-step manner beginning with development of E.F. Toro's exact Riemann solver and Godunov's first-order, upwind method. To improve computational efficiency, an adaptive Riemann solver, also introduced by Toro, replaces the exact Riemann solver. Greater stability and accuracy is achieved near steep gradients using an adaptive Courant-Friedrichs-Lewy coefficient.

Once the underlying Godunov algorithm is complete, Toro's WAF method is added to obtain second-order accuracy. Monotonicity is established

by applying one of four TVD weight limiters subsequently written into the WAF code.

To illustrate each numerical tool and its impact on computational efficiency and accuracy, a classic one-dimensional shock tube problem is evaluated over a course mesh for comparison with the solution obtained via the exact Riemann solver. While this solution is not strictly exact, because no analytical solution exists, it is suitable for use as a qualitative benchmark. Once the one-dimensional code has been tested it will be validated against experimental results obtained from the Army Research Lab's 57 cm shock tube.

Having shown the one-dimensional implementation to be sound, the code is then extended into two dimensions via the dimensional splitting scheme. To illustrate the capabilities of the code, a two-dimensional problem is solved that is representative of the types of shock problems of interest to DoD.

Chapter 2: Governing Equations and Important Thermodynamic Relations

A. Simplifying Assumptions

The basic equations of fluid motion are derived from the conservation of mass, momentum and energy. Full consideration of these conservation laws results in the Navier-Stokes equations of fluid dynamics (Anderson, 1982). These partial differential equations (PDEs) model flow with a high degree of fidelity but are complicated, computationally inefficient and impractical for many applications. The solution procedure can be simplified significantly if the following assumptions are made:

- viscosity is negligible,
- heat conduction is negligible and
- gravitational force upon air molecules is negligible.

For propagation of shocks over an ideal surface, the inertia, pressures and physical domain involved are large in comparison to the thin boundary layer and its effects along the smooth flat surface. This means that the pressure variations normal to the surface are negligible through the boundary layer and can be ignored. It is assumed then, that the pressure distribution at the surface is due entirely to the flow occurring above the boundary layer, therefore, viscosity can be ignored.

Heat conduction between the surface and air can also be ignored. As stated above, an ideal surface is flat. Another important property of our hypothetical surface is that it is perfectly reflective. Since, heat conduction can not occur across a perfectly reflective surface it can be ignored.

Lastly, viscosity and heat conduction within the air itself can be ignored. Assuming the atmosphere is homogeneous, the velocity and temperature gradients within the flow field are very small. As a result viscosity and heat conduction within the air can be neglected. Under such conditions the air is said to be adiabatic and inviscid.

An important thermodynamic property arises from these assumptions. If viscosity and heat conduction are ignored then the flow is considered to be isentropic. This means that as a particle travels through the medium, the specific entropy at the particle remains constant. The nature of the compressible flow and its interaction with the surface is now fully defined. It is inviscid and adiabatic and therefore isentropic.

One important exception occurs at the shock front where compressive forces result in significant velocity and temperature gradients. These gradients provide the necessary dissipative mechanism for the transfer of energy and momentum across the shock (Courant and Friedrichs, 1948). Here, viscosity and heat conduction effects are significant, the specific entropy is no longer constant and our simplifying assumptions are in jeopardy.

These effects are dispersive and result in a thin shock front across which the solution is smooth. However, the width of this region is roughly the mean free path of the air molecules and is microscopic when we again consider the length of the physical domain. Given the steep gradients that occur and the relative width of the shock front, we can model the shock front mathematically as a discontinuity and once again ignore viscosity and heat conduction as a good approximation. In fact, if we obtained the smooth solution using the Navier-Stokes equations and compared it with the discontinuous solution obtained via our assumptions, it would be difficult to distinguish one solution from the other (LeVeque, 1992).

It is important to note that shocks are irreversible thermodynamic processes that result in an increase in entropy, i.e. specific entropy is not constant across this region. In modeling the shock front as a discontinuity, viscosity and heat conduction effects are not ignored but are approximated in such a way that their terms can be eliminated from the governing equations. In this region, the flow is not isentropic and therefore, isentropic relations do not apply.

B. Governing Equations

Application of the assumptions made in Section A above reduces the complex Navier-Stokes equations to another coupled system of PDEs that are nonlinear and strictly hyperbolic (LeVeque, 1992). This set of equations is

commonly referred to as the Euler equations or hyperbolic conservation laws of gas dynamics.

In general vector notation, these equations are:

$$\bar{U}_t + \bar{F}(\bar{U})_x + \bar{G}(\bar{U})_y + \bar{H}(\bar{U})_z = 0. \quad (1)$$

Note that each vector above is a function of both space and time where x, y , and z are the Cartesian coordinates and t represents time. Throughout this document, subscripted independent variables indicate partial derivatives.

In Equation (1),

$$\bar{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad (2)$$

is a vector of the conserved quantities: mass, momentum (x component), momentum (y component), momentum (z component) and energy. More accurately, each vector element above represents a density function for the corresponding conserved quantity such that

$$\int U_m(x, y, z, t) dV \quad (3)$$

is the total quantity of the m^{th} conserved variable over the control volume at time t (LeVeque, 1992). In Equation (2) above, ρ is density, and u, v , and w are the components of particle velocity in the x, y and z directions respectively. E is the total energy density,

$$E = \frac{1}{2} \rho \bar{V} \cdot \bar{V} + \rho e, \quad (4)$$

where e is the specific internal energy to be defined later and \bar{V} is the particle velocity vector.

Vectors

$$\bar{F}(\bar{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ u(E + p) \end{bmatrix}, \quad \bar{G}(\bar{U}) = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho w v \\ v(E + p) \end{bmatrix} \text{ and } \bar{H}(\bar{U}) = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ w(E + p) \end{bmatrix} \quad (5)$$

represent the flux of these conserved variables in the x , y and z directions respectively.

Variables ρ, u, v, w , and p are often called primitive to distinguish them from the conserved quantities (Toro, 1997). Primitive variables are those variables that can be controlled and measured experimentally. In practice, it is convenient to transform Equation (1) into its primitive form:

$$\bar{W}_t + \bar{F}(\bar{W})_x + \bar{G}(\bar{W})_y + \bar{H}(\bar{W})_z = 0. \quad (6)$$

where \bar{W} can be expressed in terms of \bar{U} as

$$\bar{W} \equiv \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix} = \begin{pmatrix} U_1 \\ U_2/U_1 \\ U_3/U_1 \\ U_4/U_1 \\ (\gamma - 1) \left(U_5 - (0.5/U_1) (U_2^2 + U_3^2 + U_4^2) \right) \end{pmatrix}. \quad (7)$$

Equation (1) can be simplified further if we take advantage of problem symmetry and assume uniform flows accordingly. If flow in the z direction is uniform, Equation (1) reduces to the two-dimensional Euler equations,

$$\bar{U}_t + \bar{F}(\bar{U})_x + \bar{G}(\bar{U})_y = 0. \quad (8)$$

An infinite high-pressure cylinder is an example of this type and will be explored in Chapter 6. Similarly, if we assume uniform flow conditions in both y and z , reduction of Equation (1) results in the one-dimensional Euler equations,

$$\bar{U}_t + \bar{F}(\bar{U})_x = 0. \quad (9)$$

Modeling constant cross sectional area shock tubes is a classic application of Equation (9). In Chapters 3 and 4, problems of this type are explored. In either case, we are faced with a dilemma. There are $m + 1$ unknowns in m equations, so the problem is underspecified.

C. Important Thermodynamic Relations

To solve the Euler equations, an appropriate equation of state is used to derive a closure condition. Here we assume the atmosphere to be comprised of a diatomic polytropic gas where pressure is related to density and temperature by the ideal gas equation of state:

$$p = \rho R T \quad (10)$$

where R is the specific gas constant and T is the temperature. For an ideal gas, R is related to the specific heats at constant pressure and volume such that

$$R = c_p - c_v \quad (11)$$

where c_p is the specific heat at constant pressure and c_v is the specific heat at constant volume. By polytropic we mean that the specific internal energy is a function of temperature alone such that

$$e = c_v T. \quad (12)$$

Direct substitution of Equations (11) and (12) into Equation (10) yields the following important thermodynamic relation for specific internal energy, pressure and density:

$$e = \frac{P}{(\gamma - 1)\rho} \quad (13)$$

where γ is defined to be the ratio of c_p to c_v . For air at pressures below 100 psi, $\gamma \approx 1.4$.

As stated earlier in Section A, the flow is isentropic everywhere except at the shock front. This leads to the important isentropic relation

$$p \propto \rho^\gamma. \quad (14)$$

When this relation is applied in Equation (10), the following important isentropic relation is obtained:

$$\frac{\rho}{\rho_0} = \left(\frac{p}{p_0} \right)^{1/\gamma} \quad (15)$$

where the subscript zero indicates initial conditions.

Another important quantity in the study of compressible flows, and in particular flows where shocks occur, is the local speed of sound, c , where

$$c^2 = \left(\frac{\partial p}{\partial \rho} \right)_s. \quad (16)$$

Differentiation of Equation (14) yields the local speed of sound for a polytropic gas given constant entropy:

$$c = \sqrt{\frac{\gamma p}{\rho}}. \quad (17)$$

At this point we now have all the principle equations necessary to build our numerical methods and solve shock problems. In the next chapter, we introduce and solve an initial value problem (IVP) of importance called the Riemann problem. As we shall see, this IVP forms the backbone of all our numerical techniques.

Chapter 3: The Riemann Problem and Its Solution

A. The Riemann Problem

The Riemann problem for the one-dimensional Euler equations is the IVP with initial conditions (Chang and Hsiao, 1989):

$$\bar{U}(x,0) = \begin{cases} \bar{U}_1(x,0) & x < 0, \\ \bar{U}_4(x,0) & x > 0. \end{cases} \quad (18)$$

The domain is centered about the discontinuity at $x = 0m$ and includes all points (x,t) in the $x-t$ plane such that $-\infty < x < \infty$ and $t > 0s$. In practice, the domain is restricted about a finite spatial interval, $-\Delta x/2 \leq x \leq \Delta x/2$.

An important mathematical property of the solution is its self-similarity nature (LeVeque, 1992; AGARD, 1961). Provided the initial discontinuity is located at $x = 0m$, the solution is a function of one variable, $\zeta = x/t$, such that

$$\bar{U}(x,t) = \bar{U}(\zeta). \quad (19)$$

Given Equation (19), the solution is said to be *similar* at different times along characteristic curves, where $\zeta = \text{constant}$, in the $x-t$ plane. It follows that because $\zeta = \text{constant}$, $dx/dt = \text{constant}$ so the solution propagates along these characteristic curves at a constant speed as well.

This self-similarity property forms the basis of a powerful technique used to solve linear first-order hyperbolic PDEs called the method of characteristics (AGARD, 1961; Abbott, 1966; Anderson, 1982). In the method

of characteristics, the solution to a linear hyperbolic system of m PDEs is obtained by solving two sets of m ordinary differential equations (ODEs): set one defines the characteristic curves along which the solution remains constant and propagates and set two defines compatibility equations which hold true along these characteristics. Unfortunately, the Euler equations are nonlinear resulting in discontinuities where characteristics cross, so the solution procedure is not so direct. Despite this, much of the characteristic information obtained via the method of characteristics still remains useful.

B. General Wave Structure of the Riemann Problem

The method of characteristics plays an important role in defining the general wave structure of the Riemann problem. Using the method of characteristics, we obtain three characteristic fields, C^- , C^0 , and C^+ with characteristic speeds (AGARD, 1961; Toro, 1997):

$$\lambda_1 = u - c, \quad (20)$$

$$\lambda_2 = u \quad (21)$$

and

$$\lambda_3 = u + c \quad (22)$$

respectively. The superscripts $-$, 0 , and $+$ reflect the role of c , the local speed of sound, in the above equations.

Each characteristic field presented above defines a wave in the RP domain such that wave 1, is defined by C^- characteristics, wave 2 by

C^0 characteristics and wave 3 by C^+ characteristics. In the $x-t$ plane, these waves partition the RP domain into four regions as shown in Figure 1:

R_1, R_2, R_3 and R_4 where

$$\bar{W}(x,t) = \begin{pmatrix} \bar{W}_1 = (\rho_1, u_1, p_1)^T & \text{in } R_1 \\ \bar{W}_2 = (\rho_2, u_2, p_2)^T & \text{in } R_2 \\ \bar{W}_3 = (\rho_3, u_3, p_3)^T & \text{in } R_3 \\ \bar{W}_4 = (\rho_4, u_4, p_4)^T & \text{in } R_4 \end{pmatrix}. \quad (23)$$

$\bar{U}(x,t)$ has been recast into its primitive variable form using Equation (7) for convenience. Throughout most of our discussion and computations, the primitive variable form can and will be used.

The outer-waves, 1 and 3, can be either rarefactions or shocks depending upon their characteristic fields. If the characteristic field is divergent the wave is a rarefaction; if it is convergent the wave is a shock (see Figures 2 and 3). Until the true nature of the wave is determined, each wave is represented by a pair of rays meant to represent the head and tail characteristics of a rarefaction. The inner-wave, wave 2, is always a contact surface and is represented by one ray. The large arrows in Figure 1 show the motion of these waves.

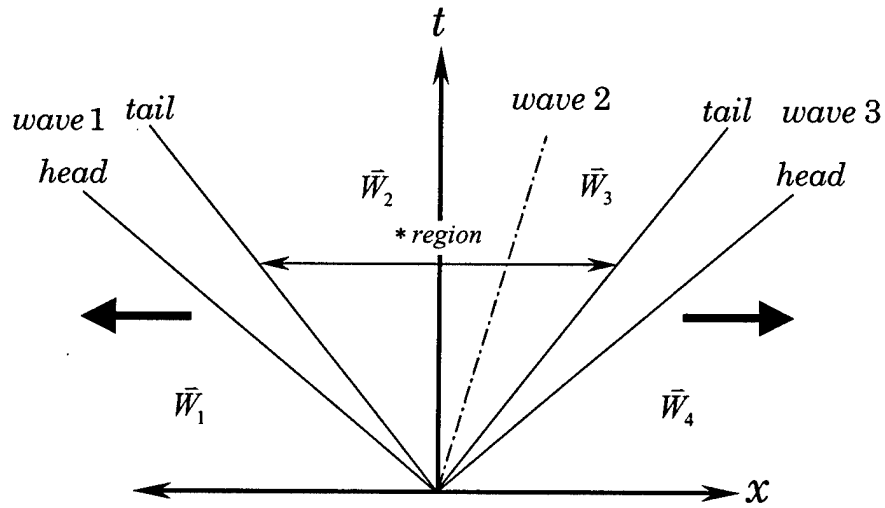


Figure 1: General Wave Structure of the Riemann Problem

If the outer wave is a rarefaction, the solution varies continuously along divergent characteristics that lie between the head and tail. For example, we see that the C^0 characteristic field defines a left traveling rarefaction wave as depicted in Figure 2. Because the characteristic field is

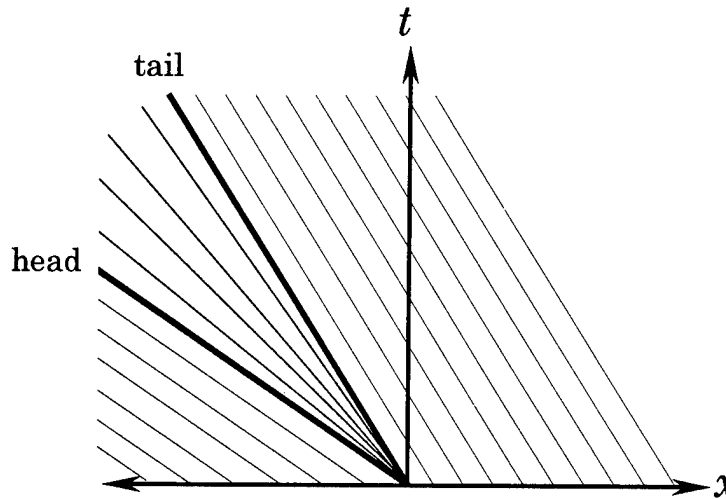


Figure 2: Characteristics Defining a Left Traveling Rarefaction

divergent, pressure and density decrease as we travel across the expansion fan from head to tail. The particle velocity may increase or decrease depending upon whether the rarefaction is traveling to the left or right. If wave 1 is a rarefaction, particle velocity increases. If wave 3 is a rarefaction, particle velocity decreases. The speed of the rarefaction wave is taken to be that of the head characteristic.

If the outer wave is a shock, characteristics converge and cross as shown in Figure 3 for a right traveling shock. The solution is discontinuous

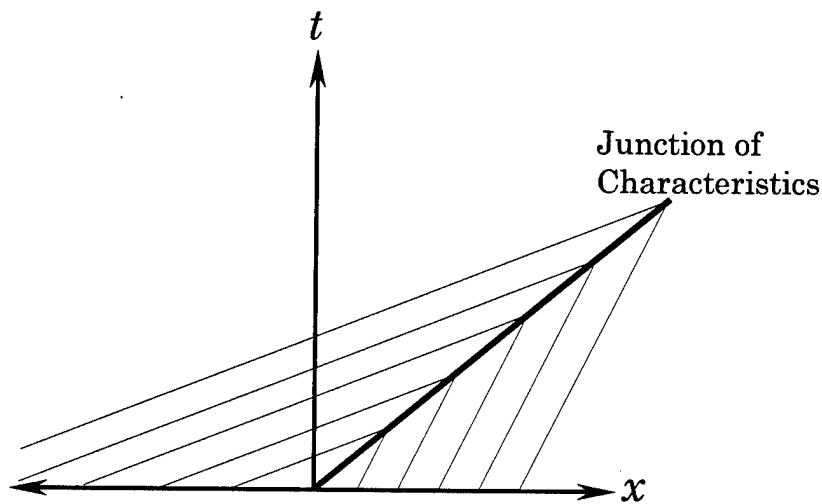


Figure 3: Characteristics Defining a Right Traveling Shock

at the point where they meet. By convention, the head and tail are dropped and the shock front is represented by a bold solid line that represents points where these characteristics cross. Each primitive variable experiences an instantaneous jump across this line as we cross the pair of rays from head to tail. The speed of the shock wave is not that of the head characteristic since

this wave itself is not a characteristic. The speed of the shock wave will be introduced later.

As stated above, the middle wave is always a contact surface. Here, characteristics run parallel to the contact surface as shown for a right traveling contact surface in Figure 4. Particle velocity and pressure are

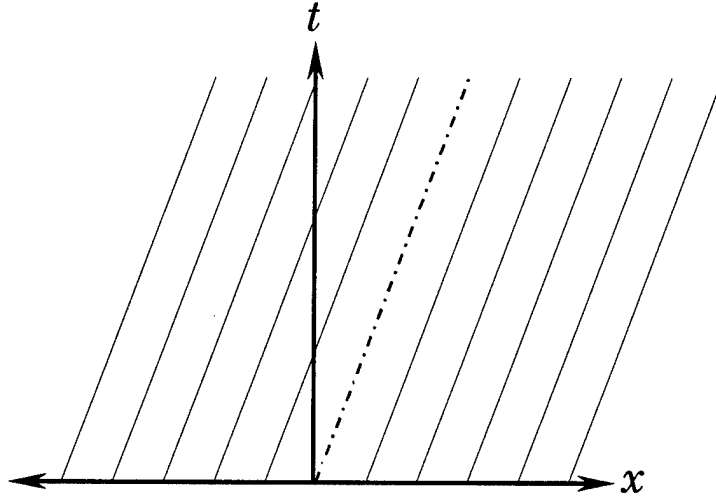


Figure 4: Characteristics Defining a Right Traveling Contact Surface

constant across the contact surface, i.e.

$$u_* \equiv u_2 = u_3 \quad (24)$$

and

$$p_* \equiv p_2 = p_3. \quad (25)$$

Because there is no pressure differential across wave 2 and particle velocities are the same on either side, the flow remains separated resulting in a discontinuity in density. The region that lies between waves 1 and 3 is called the star region, R_* . Note that R_* consists of R_2 and R_3 .

The complicated nature of the waves described above means the solution of the Riemann problem can have four possible wave configurations as shown in Table 1. R, C and S

wave 1	wave 2	wave 3
R	C	S
S	C	R
R	C	R
S	C	S

Table 1: Possible Wave Configurations

indicate whether the wave is a rarefaction, contact surface or shock. Two-rarefaction and two-shock configurations can occur when waves interact with each other or reflect at boundaries.

C. Important Wave Relations

Recall that \bar{W}_1 and \bar{W}_4 are given by Equation (7). To solve the Riemann problem we must find the unknown states, \bar{W}_2 and \bar{W}_3 , that lie within R_* . Algebraic relations are derived from physical principles in any fluid dynamics text and can be used to connect known states to the unknown states, \bar{W}_2 and \bar{W}_3 , across waves 1 and 3 respectively (Anderson, 1982; Courant and Friedrichs, 1948). E.F. Toro has developed relations that connect these states via the following particle velocity equations (1989):

$$u_2 = u_1 - f_1(p_*, \bar{W}_1) \quad (26)$$

and

$$u_3 = u_4 + f_4(p_*, \bar{W}_4). \quad (27)$$

Recall that the outer waves can be either rarefactions or shocks. This means the definition of the pressure functions, f_1 and f_4 , is dependent upon the wave types. If the outer wave is a shock, the Rankine-Hugoniot relations are used to derive these functions. If the wave is a rarefaction, the compatibility equations, called Riemann invariants, along the C^- and C^+ characteristics and isentropic relations are used. Application of these relations results in the following generalized pressure functions (Toro, 1997):

$$f_\xi(p_*, \bar{W}_\xi) = \begin{cases} (p_* - p_\xi) \left(\frac{2}{\rho_\xi(p_*(\gamma+1) + p_\xi(\gamma-1))} \right)^{1/2} & \text{if } p_* > p_\xi \quad \text{shock} \\ \frac{2c_\xi}{\gamma-1} \left(\left(\frac{p_*}{p_\xi} \right)^{\gamma-1/2\gamma} - 1 \right) & \text{if } p_* \leq p_\xi \quad \text{rarefaction} \end{cases} \quad (28)$$

where ξ represents the known constant state, 1 or 4.

Within R_* , u_* is constant, that is

$$u_3 - u_2 = 0. \quad (29)$$

Substitution of Equations (26) and (27) into Equation (29) yields Toro's nonlinear algebraic pressure function (Toro, 1997):

$$f(p_*, \bar{W}_1, \bar{W}_4) \equiv f_1(p_*, \bar{W}_1) + f_4(p_*, \bar{W}_4) + u_4 - u_1 = 0. \quad (30)$$

Given constant states 1 and 4, p_* is found numerically using a root solver.

Once p_* is found, the other state variables follow directly. The particle velocity within the star region can be calculated using either Equation (26) or (27). However, because p_* is a numerical approximation, albeit a rather accurate one, obtained using a root solver, u_* is calculated as (Toro, 1997):

$$u_* = \frac{1}{2}(u_1 + u_4) + \frac{1}{2}(f_4 - f_1). \quad (31)$$

Unfortunately, density within R_2 and R_3 is dependent upon the wave types of the outer waves and therefore no simple relation exists to calculate these variables. We must again use a generalized density function derived from the same relations used above to derive Equation (28). To calculate ρ_2 and ρ_3 , the following generalized function is used (Toro, 1997):

$$d_{\xi \pm 1}(p_*, \rho_\xi, p_\xi) = \begin{cases} \rho_\xi \left(\frac{p_*(\gamma + 1) + p_\xi(\gamma - 1)}{p_*(\gamma - 1) + p_\xi(\gamma + 1)} \right) & \text{if } p_* > p_\xi \quad \text{shock} \\ \rho_\xi \left(\frac{p_*}{p_\xi} \right)^{1/\gamma} & \text{if } p_* \leq p_\xi \quad \text{rarefaction} \end{cases} \quad (32)$$

where ξ again denotes variables from the known constant states.

D. Toro's Exact Riemann Solver (ERS)

With these relations now in hand, we seek the exact solution to the one-dimensional Euler equations given the initial conditions in Equation (18). Unfortunately, no closed form solution exists. We can however, approximate the solution with as high a degree of accuracy as our

computational resources will allow. To find the exact solution, we have chosen Toro's exact Riemann solver for its simplicity and computational efficiency (Toro, 1989).

The exact Riemann solver consists of three principle steps, as shown in Figure 5, beginning with the most computationally expensive portion of the

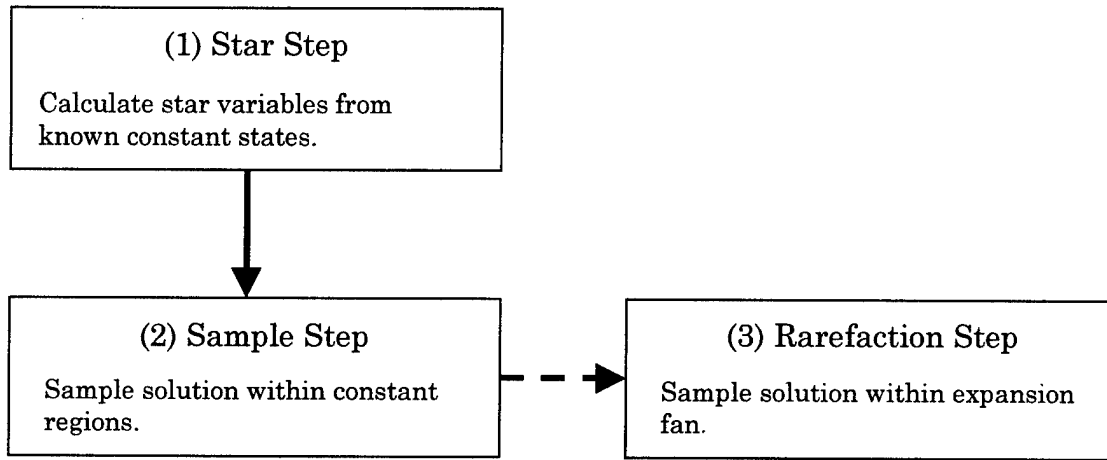


Figure 5: Exact Riemann Solver Solution Procedure

entire algorithm, the star step. In this step, the variables within R_* are obtained given known states \bar{W}_1 and \bar{W}_4 using the wave relations presented in Section C. Once \bar{W}_2 and \bar{W}_3 are known, the solution is sampled at a user-defined number of points. This is called the sample step. If, during the star step, one of the outer waves is identified as a rarefaction, the sample points within the expansion fan are determined using another set of equations.

The Star Step

Toro's exact Riemann solver begins with the star step. Notice from Equations (31) and (32) that u_* , ρ_2 and ρ_3 are all functions of the

remaining star variable, p_* . This means p_* must be found before the solution can be obtained fully. The pressure within the star region, p_* , is approximated by finding the root of Equation (30) iteratively using the Newton-Raphson method. This method takes the form (Burden and Faires, 1993):

$$p_*^{(k)} = p_*^{(k-1)} - \frac{f(p_*^{(k-1)}, \bar{W}_1, \bar{W}_4)}{\frac{\partial}{\partial p_*} f(p_*^{(k-1)}, \bar{W}_1, \bar{W}_4)} \quad (33)$$

where k denotes the k^{th} iteration. It is important to reiterate here the fact that both pressure functions, f_1 and f_4 , are defined differently depending upon the nature of the outer-waves. Therefore, if we wish to converge rapidly to the solution, waves 1 and 3 must be first be characterized and then the appropriate pressure function applied.

Given $f' \neq 0$ and a sufficiently close initial approximation, $p_*^{(0)}$, Equation (33) will converge quadratically to the root. To find an adequate initial guess, Toro's adaptive initial guess generator is used (see Appendix A). The resulting approximation is considered adequate when the relative error is less than the user-defined error, ε , (Toro, 1989):

$$\frac{|p_*^{(k)} - p_*^{(k-1)}|}{\frac{1}{2}(p_*^{(k)} + p_*^{(k-1)})} < \varepsilon. \quad (34)$$

After p_* is found, the remaining star variables are calculated per Section C and the solution within constant regions 1 through 4 is fully obtained. Now the solution is sampled across the computational domain.

The Sample Step

In the sample step, we sample the solution at a user-defined number of points. The solution at (x, t) is obtained by exploiting the self-similar nature and wave structure of the Riemann problem. Recall that for the solution to be self-similar, the Riemann problem domain must be symmetric about the initial discontinuity located at $\hat{x} = 0$ as shown in Figure 6. S_1 , S_2 and S_3 represent the speeds of waves 1, 2 and 3 respectively. In the laboratory

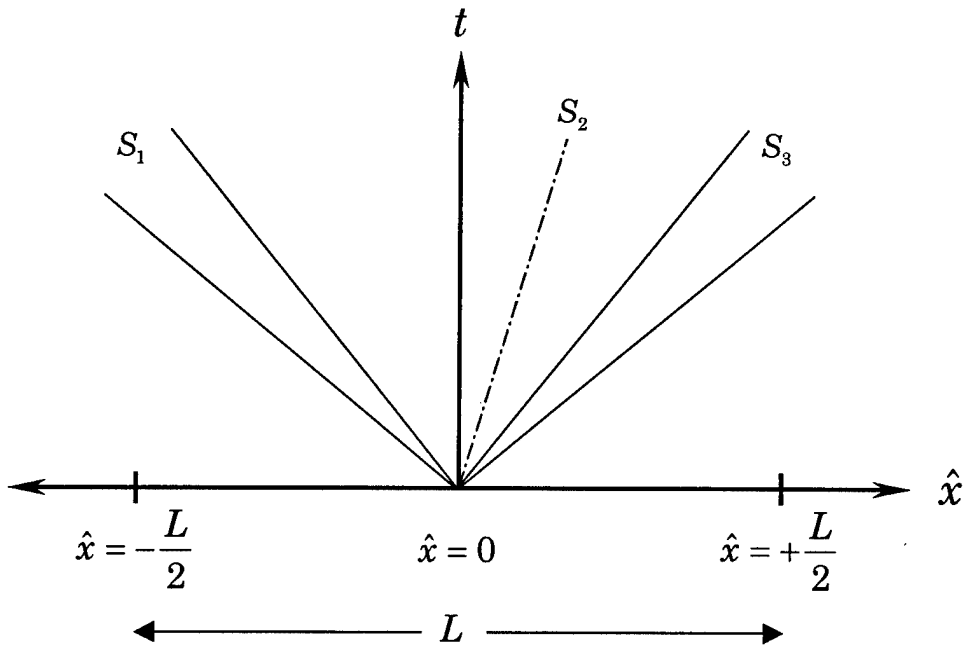


Figure 6: Riemann Problem Local Reference Frame

reference frame the discontinuity is located at $x = L/2$ where L represents the physical length of the problem. The physical domain is typically $[0, L] \times [0, T]$ where T is the desired solution time. To transform from the physical to computational domain, the following coordinate transformation must be performed to meet the conditions required for similarity:

$$\hat{x} = x - \frac{L}{2} \quad (35)$$

where \hat{x} and x represent position in the local and laboratory reference frames respectively.

Now we take advantage of the similarity properties described in Section B to determine $\bar{W}(x, T)$. Here we define

$$\hat{S} = \frac{\hat{x}}{T} \quad (36)$$

where \hat{S} is the speed required to move to \hat{x} at the time T starting at $(0m, 0s)$ in the Riemann problem reference frame.

Given \hat{S} , the solution is easily obtained by comparing it with the wave speeds. If no rarefactions are present, the solution is:

$$\bar{W}(x, T) = \begin{cases} \bar{W}_1 & \text{if } \hat{S} \leq S_1 \\ \bar{W}_2 & \text{if } S_1 < \hat{S} < S_2 \\ \bar{W}_3 & \text{if } S_2 < \hat{S} < S_3 \\ \bar{W}_4 & \text{if } \hat{S} > S_4 \end{cases} \quad (37)$$

where the wave speeds are (Toro, 1991):

$$S_1 = \begin{cases} u_1 - c_1 \left[\frac{\gamma+1}{2\gamma} \frac{p_*}{p_1} + \frac{\gamma-1}{2\gamma} \right]^{1/2} & \text{if } p_* > p_1 \quad \text{shock} \\ u_1 - c_1 & \text{if } p_* \leq p_1 \quad \text{rarefaction} \end{cases}, \quad (38)$$

$$S_2 = u_*, \quad (39)$$

and

$$S_3 = \begin{cases} u_4 + c_4 \left[\frac{\gamma+1}{2\gamma} \frac{p_*}{p_4} + \frac{\gamma-1}{2\gamma} \right]^{1/2} & \text{if } p_* > p_4 \quad \text{shock} \\ u_4 + c_4 & \text{if } p_* \leq p_4 \quad \text{rarefaction} \end{cases}. \quad (40)$$

If a rarefaction is present, the solution within the expansion fan is calculated using a different set of equations. This occurs in the rarefaction step.

The Rarefaction Step

As stated earlier, if a rarefaction is present special care must be taken because the solution varies continuously between the head and tail of the expansion fan. Figure 7 depicts a typical solution of the Riemann problem that consists of a left traveling rarefaction, right traveling contact surface

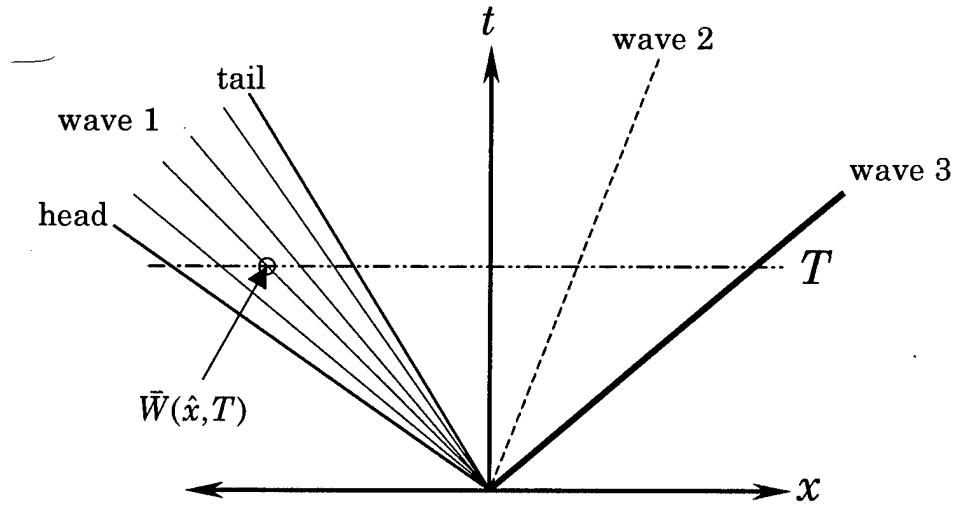


Figure 7: Typical Wave Structure

and a right traveling shock. The solution is desired at point (\hat{x}, T) . We determine it lies within the expansion fan using the following relation:

$$S_1^{head} < \hat{S} < S_1^{tail} \quad (41)$$

where

$$S_1^{head} = u_1 - c_1 \quad (42)$$

and

$$S_1^{tail} = u_* - c_2. \quad (43)$$

To determine the solution within a left rarefaction, the following set of equations is used (Toro, 1997):

$$\bar{W}_L^{fan}(\hat{S}) = \begin{pmatrix} \rho = \rho_1 \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{(\gamma+1)c_1} (u_1 - \hat{S}) \right]^{\frac{2}{\gamma-1}} \\ u = \frac{2}{\gamma+1} \left(c_1 + \frac{\gamma-1}{2} u_1 + \hat{S} \right) \\ p = p_1 \left[\frac{2}{\gamma+1} + \frac{\gamma-1}{(\gamma+1)c_1} (u_1 - \hat{S}) \right]^{\frac{2\gamma}{\gamma-1}} \end{pmatrix}. \quad (44)$$

Similarly, we use the following relations to determine if (\hat{x}, T) lies within a right rarefaction (Toro, 1997).

$$S_3^{tail} < \hat{S} < S_3^{head} \quad (45)$$

$$S_3^{head} = u_4 + c_4 \quad (46)$$

$$S_3^{tail} = u_* + c_3. \quad (47)$$

If a right rarefaction is present then the following set applies (Toro, 1997):

$$\bar{W}_R^{fan}(\hat{S}) = \begin{pmatrix} \rho = \rho_4 \left[\frac{2}{\gamma+1} - \frac{\gamma-1}{(\gamma+1)c_4} (u_4 - \hat{S}) \right]^{\frac{2}{\gamma-1}} \\ u = \frac{2}{\gamma+1} \left(-c_4 + \frac{\gamma-1}{2} u_4 + \hat{S} \right) \\ p = p_4 \left[\frac{2}{\gamma+1} - \frac{\gamma-1}{(\gamma+1)c_4} (u_4 - \hat{S}) \right]^{\frac{2\gamma}{\gamma-1}} \end{pmatrix}. \quad (48)$$

The entire solution of the Riemann problem is now defined.

E. The Sod Shock Tube Problem

To illustrate a typical solution to the Riemann problem, a mild shock tube problem is solved using the exact Riemann solver with 10001 points

(Figures 8-11). The solution consists of a left rarefaction, a contact and a shock on the right. The wave structure is similar to that depicted in Figure 7. The shock tube is one meter long with a diaphragm positioned at $x = 0.5m$. The following initial conditions exist:

$$\bar{W}(x,0) = \begin{cases} \bar{W}_1 = (1.000 \text{ kg/m}^3, 0.0 \text{ m/s}, 1.000 \text{ Pa})^T & x < 0.5m \\ \bar{W}_4 = (0.125 \text{ kg/m}^3, 0.0 \text{ m/s}, 0.100 \text{ Pa})^T & x > 0.5m \end{cases} \quad (49)$$

This problem was first presented and used by Sod (Sod, 1978).

At $t = 0s$, the diaphragm is instantaneously removed resulting in particle flow from left to right down the pressure gradient. As this flow continues it accelerates such that particle velocities become supersonic relative to the ambient conditions or quiet conditions downwind (Figure 9). As a result, compression waves coalesce forming a shock that travels to the right into the low-pressure region (Figure 10). Directly behind the shock, a contact surface follows traveling at a lower speed (Figure 8). The contact surface occurs because the pressure and particle velocity behind the shock is constant within R_* . This results in a difference in internal energy between the colder air to the left of the contact discontinuity and the shock-heated air to the right as observed in Figure 11. In the opposite direction, a rarefaction wave travels to the left into the high-pressure region. Notice in Figure 9 that the particle velocity is positive and increases across the rarefaction wave. The expansion fan occurs due to the low density region created locally as the

heated gas moves rapidly to the right down the pressure gradient. This results in a region where pressure, density and temperature decrease.

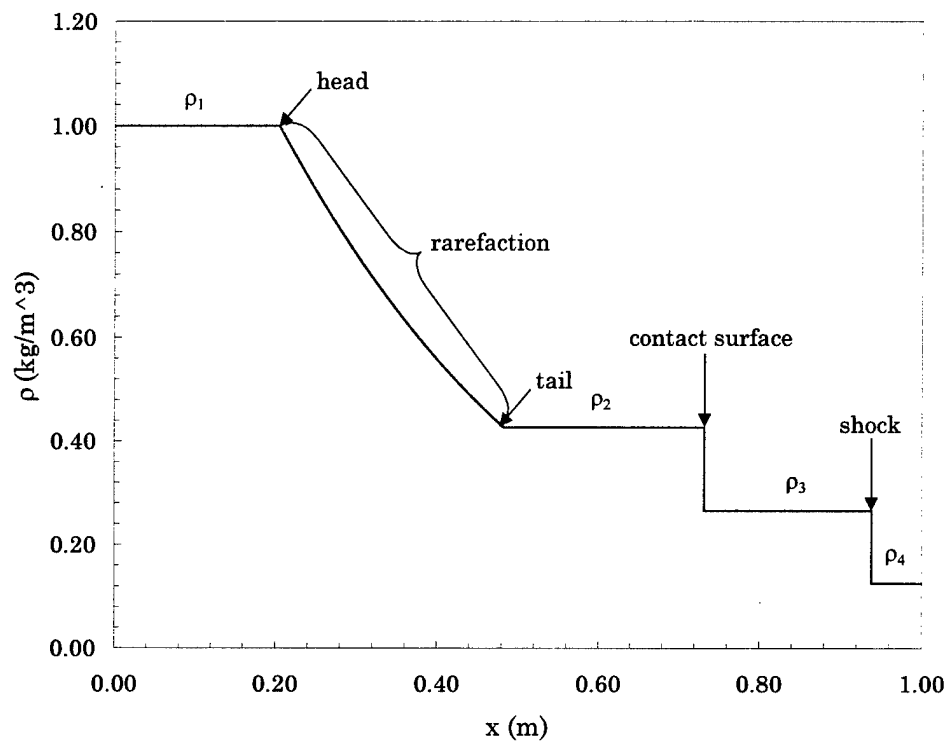


Figure 8: Density Profile at $t=0.25$ s

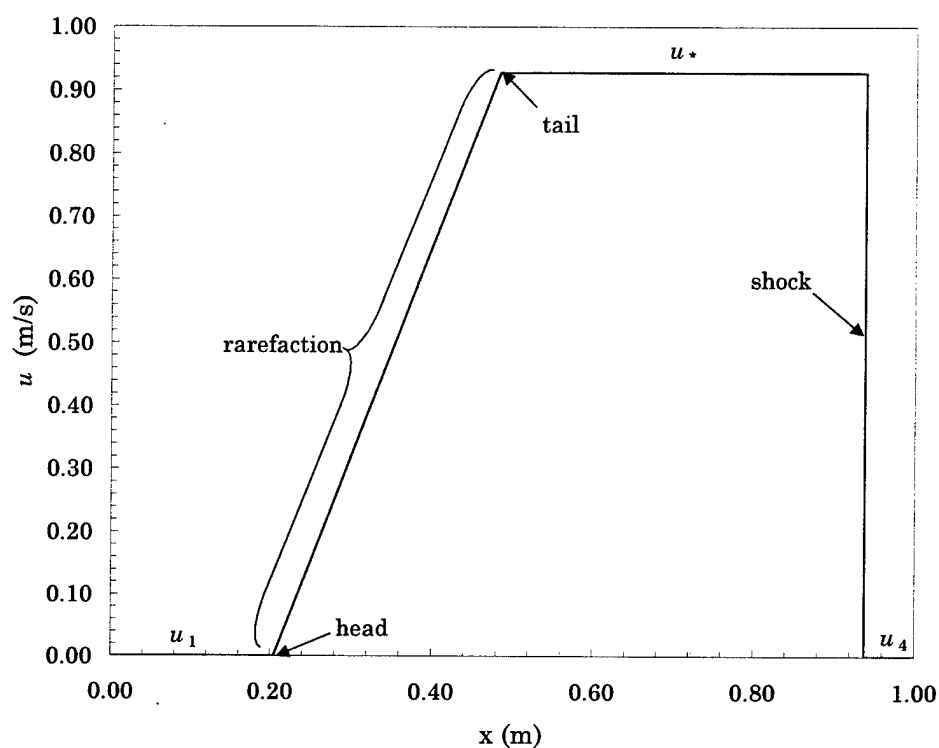


Figure 9: Particle Velocity Profile at $t=0.25s$

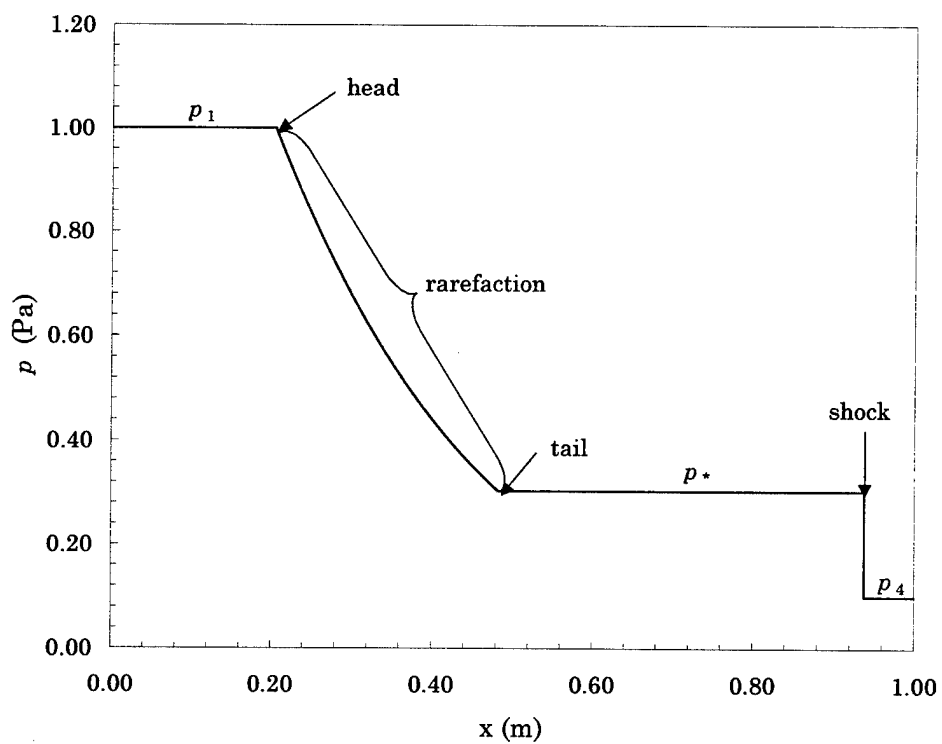


Figure 10: Pressure Profile at $t=0.25s$

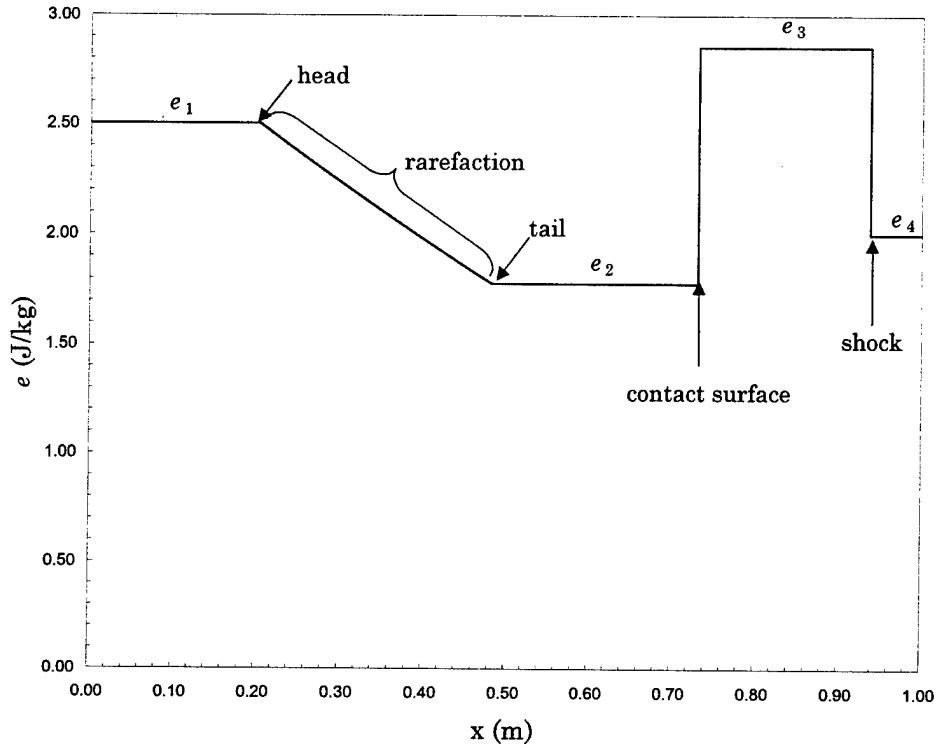


Figure 11: Specific Internal Energy Profile at $t=0.25s$

To test the implementation of the exact Riemann solver, we solved five shock tube problems in addition to Sod's:

- Toro's shock tube problem (1991),
- two-rarefaction shock tube problem (Einfeldt and others, 1991),
- left half of Woodward and Colella's blast problem (1984),
- right half of Woodward and Colella's blast problem (1984) and
- two-shock shock tube problem (Toro, 1997).

In each case, we found the results to be in agreement with those presented in the literature.

Chapter 4: One-dimensional Shock Code Implementation

While the exact Riemann solver provides an exact solution unfortunately, it solves only the simplest of problems, namely symmetric one-dimensional shock tube type problems. To solve more interesting problems we must turn to finite volume techniques. Here we describe, implement and test several numerical techniques to build a second-order accurate, one-dimensional shock code.

A. Initial Boundary Value Problem

In this chapter we will consider and subsequently solve the following general initial boundary value problem (IBVP):

$$IBVP = \begin{cases} \bar{U}_t + \bar{F}(\bar{U})_x = \bar{0} \\ \bar{U}(x,0) = \bar{U}^0(x) \end{cases} \quad (50)$$

Unlike the RP described in the last chapter, no symmetry is assumed. In the most general sense, $\bar{U}^0(x)$ is a piecewise function that defines multiple discontinuities at $t = 0s$ over the spatial domain, $[0, L]$. The boundary conditions along $x = 0m$, and $x = L$ are explicitly defined to be either transmissive or reflective. These boundary conditions arise from the numerical requirement to model flows along physical and computational boundaries.

B. Discretization of the Domain

To solve Equation (50), the domain, $[0, L] \times [0, T]$, is discretized in the $x-t$ plane forming a rectilinear mesh as shown in Figure 12. Mesh indices i and n represent the i^{th} cell at the n^{th} time level. The spatial domain is

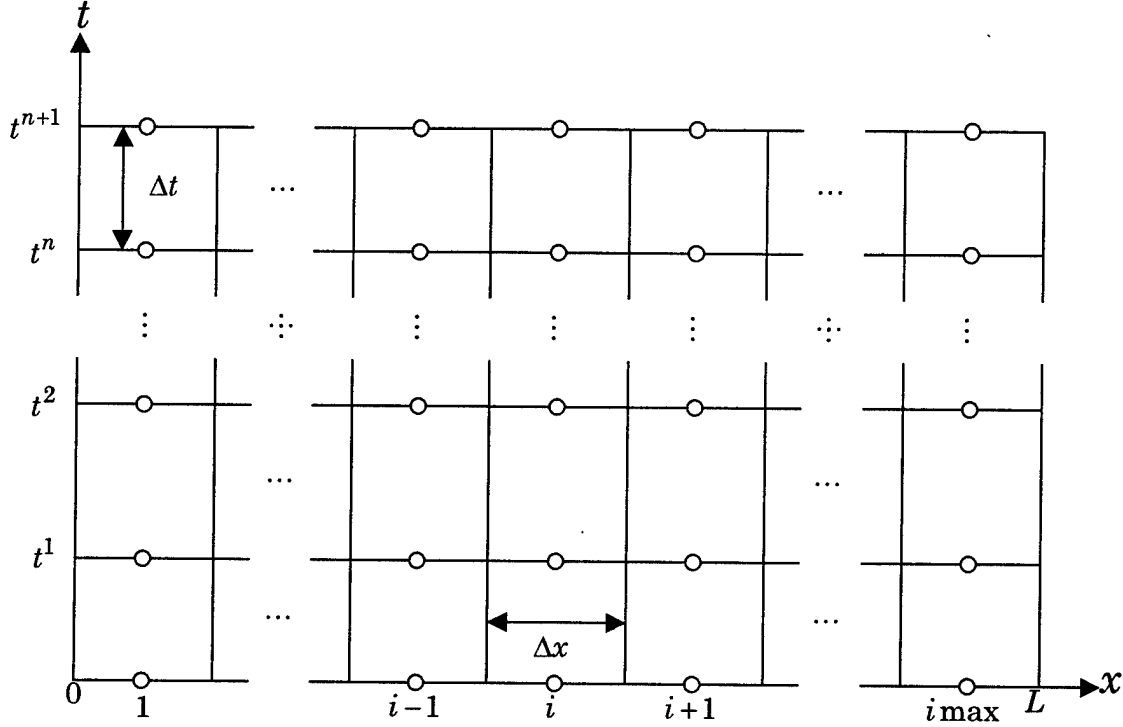


Figure 12: Discretized Domain

subdivided into I computational cells of uniform width Δx such that:

$$\Delta x = \frac{L}{I}. \quad (51)$$

The center of the i^{th} computational cell is located at:

$$x_i = L \left(\frac{i - 0.5}{I} \right) \text{ for } i = 1, 2, \dots, I. \quad (52)$$

The extent of the i^{th} cell is defined by faces $i - 1/2$ and $i + 1/2$ located at

$$x_{i-1/2} = L \left(\frac{i-1}{I} \right) \quad (53)$$

and

$$x_{i+1/2} = L \left(\frac{i}{I} \right) \quad (54)$$

where

$$\Delta x = x_{i+1/2} - x_{i-1/2}. \quad (55)$$

Notice in Figure 12 that each time step, Δt , is not constant but determined adaptively (Toro, 1997):

$$\Delta t^n = \frac{C \Delta x}{\tilde{S}_{\max}^n} \text{ for } n = 0, 1, \dots, N. \quad (56)$$

where C is called the Courant coefficient and \tilde{S}_{\max}^n is the estimated maximum wave speed across the spatial domain at time t^n .

We now have a computational mesh that is fully defined with mesh points located at the center of each computational cell at every time level. As we numerically solve Equation (50), cell-averaged values of the solution, $\bar{U}(x, t)$, are assigned to mesh points and are denoted:

$$\bar{U}_i^n \equiv \bar{U}(x_i, t^n). \quad (57)$$

C. Conservative Discretization of the Euler Equations

Recall from Chapter 3 that the solution to the Euler equations is not smooth everywhere. We have shown that discontinuities occur at the shock front and contact surface. In these regions, the solution is not differentiable

and the differential form of the Euler equations fails. The PDEs are integrable over these same discontinuous regions so we recast the PDEs into the following form:

$$\oint [\bar{U} dx - \bar{F}(\bar{U}) dt] = 0. \quad (58)$$

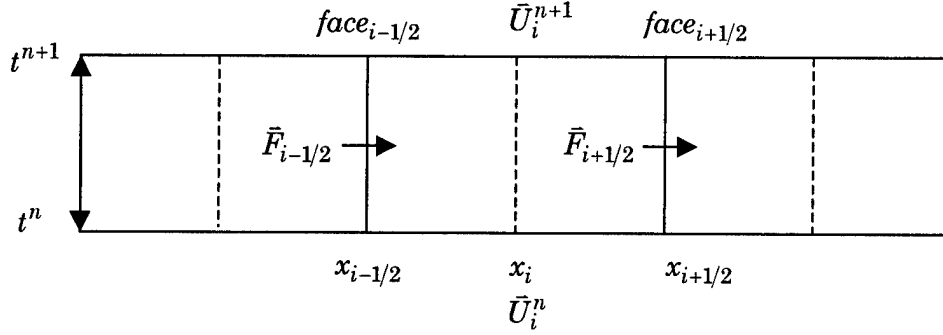


Figure 13: Arbitrary Mesh Rectangle

If we apply Green's theorem in the $x-t$ plane and integrate Equation (58) over an arbitrary rectangle, shown in Figure 13, the conservative discretized form is obtained:

$$\bar{U}_i^{n+1} = \bar{U}_i^n + \frac{\Delta t}{\Delta x} [\bar{F}_{i-1/2} - \bar{F}_{i+1/2}]. \quad (59)$$

\bar{U}_i^{n+1} and \bar{U}_i^n are space-integrated averages of the solution over cell i where

$$\bar{U}_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{U}(x, t^{n+1}) dx \quad (60)$$

and

$$\bar{U}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{U}(x, t^n) dx. \quad (61)$$

Similarly, $\bar{F}_{i-1/2}$ and $\bar{F}_{i+1/2}$, represent the time-integrated averages of the physical flux of \bar{U} through cell faces $i-1/2$ and $i+1/2$ over time step Δt .

These flux functions are obtained in the following manner:

$$\bar{F}_{i-1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \bar{F}(\bar{U}(x_{i-1/2}, t)) dt \quad (62)$$

and

$$\bar{F}_{i+1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \bar{F}(\bar{U}(x_{i+1/2}, t)) dt. \quad (63)$$

Equation (59) is a direct mathematical statement of the conservation laws. It describes the variation in cell-averaged variables over Δt which result from the balance of the time-averaged fluxes at the cell faces. The solution procedure is obvious. \bar{U}_i^{n+1} is calculated from \bar{U}_i^n and the net flux, $(\bar{F}_{i-1/2} - \bar{F}_{i+1/2})$, through cell i . Unfortunately, $\bar{F}_{i-1/2}$ and $\bar{F}_{i+1/2}$ are not known a priori. Therefore, we need a numerical scheme that approximates the flux through cell i given \bar{U}_{i-1}^n and \bar{U}_i^n .

D. Godunov's Method

To solve the IBVP, we use Godunov's conservative upwind scheme (LeVeque, 1992; Hirsch, 1990). This method assumes the solution, $\bar{U}(x, t)$, has a piecewise constant distribution of cell averages over the spatial domain at each time level. At $t = 0s$, we map $\bar{U}^0(x)$ onto our computational domain using cell-averaged values as defined by Equation (61). For example, let's

assume a discontinuity exists at $x = x_3$ as depicted in Figure 14. Then the cell-averaged value assigned in cell 3 is simply $\bar{U}_3^0 = (\bar{U}_2^0 + \bar{U}_4^0)/2$.

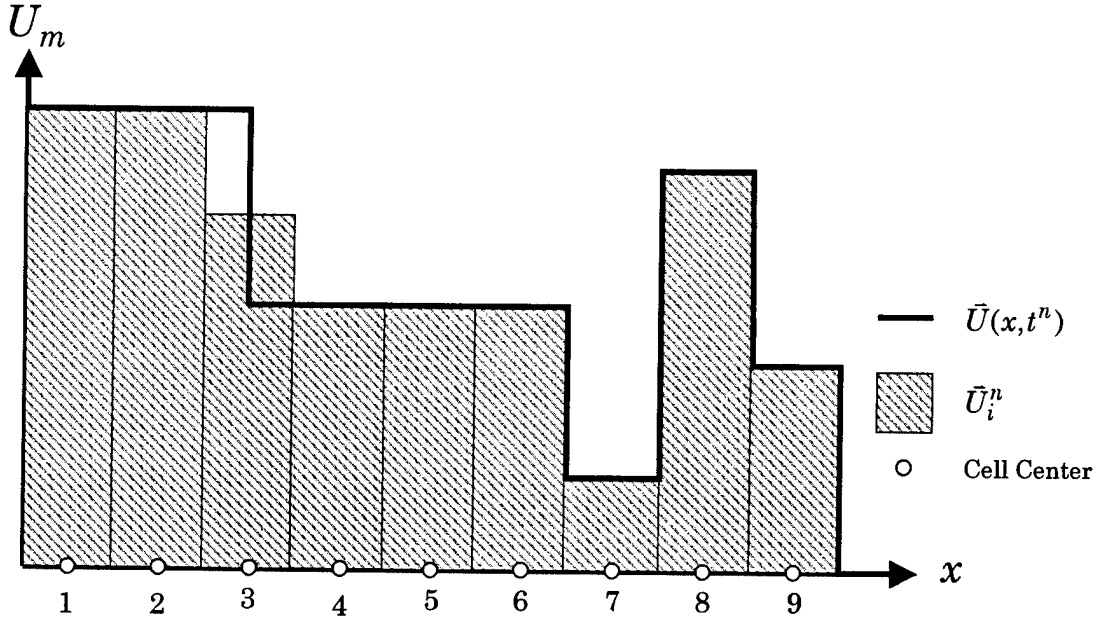


Figure 14: m^{th} Component of $\bar{U}(x, t^n)$ Mapped onto Discretized Spatial Domain

At this point, we have defined a set of I constant states that may be interpreted as $I - 1$ pairs, $\{\bar{U}_i^n, \bar{U}_{i+1}^n\}$. Each of these pairs is separated by a discontinuity that occurs at the shared cell face. Recall $\bar{F}_{i-1/2}$ and $\bar{F}_{i+1/2}$ are not known but are required if the solution at t^{n+1} is to be found.

Fortunately, our interpretation of the solution distribution suggests a simple procedure. We can calculate the unknown fluxes by defining and solving a series of local Riemann problems exactly at each cell face:

$$IVP = \begin{cases} \bar{U}_t + \bar{F}(\bar{U})_x = 0 \\ \bar{U}(x,0) \end{cases} \quad (64)$$

where

$$\bar{U}(x,0) = \begin{cases} \bar{U}_i & x < 0 \\ \bar{U}_{i+1} & x > 0 \end{cases}. \quad (65)$$

Recall that, in practice, we solve the Riemann problem in primitive variable form as stated in Chapter 3. From the exact solution of Equation (64), we obtain the constant solution along face $i+1/2$. Evaluation of Equations (62) and (63) is now trivial. In general, the Godunov flux is:

$$\bar{F}_{i+1/2}^{GF} = \bar{F}(\bar{W}(x_{i+1/2})) \quad (66)$$

where $\bar{W}(x_{i+1/2})$ is dependent upon position only and represents the primitive variable state along face $i+1/2$. Once $\bar{F}_{i-1/2}$ and $\bar{F}_{i+1/2}$ are known, we obtain \bar{U}_i^{n+1} using Equation (59).

E. Implementation of Godunov's Method

To calculate the solution at t^{n+1} from t^n , we implement Godunov's method using six steps as shown in Figure 15. In step 1, boundary conditions are applied to fictitious cells, 0 and $I+1$, called phantom cells. As we shall see, these cells are required to obtain the solution in border cells 1 and I . Once boundary conditions have been applied, Δt is determined subject to the Courant-Friedrichs-Lewy stability condition.

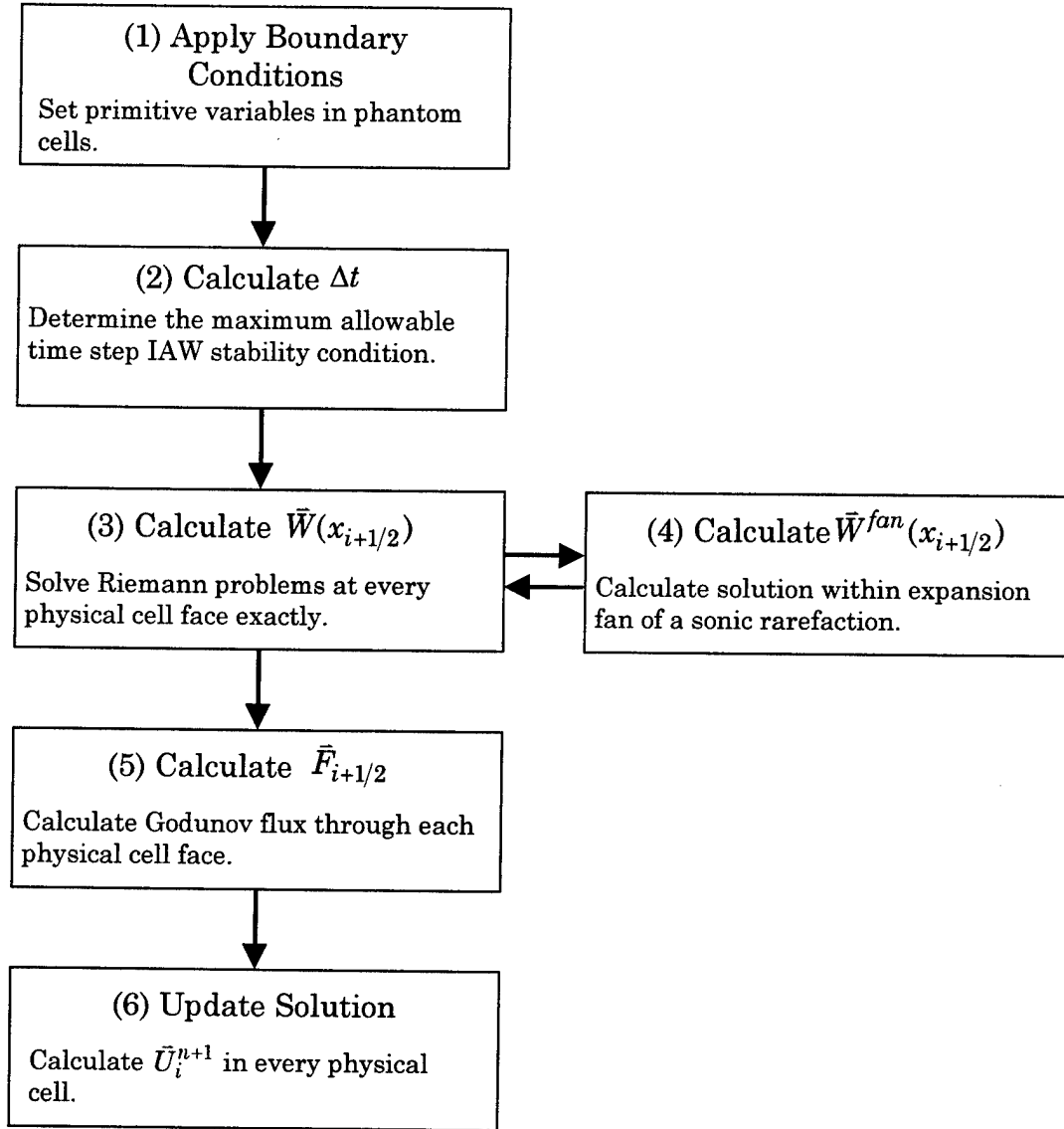


Figure 15: Godunov Solution Procedure

In step 3, we solve all $I + 1$ RPs to obtain $\bar{W}(x_{i+1/2})$ along each physical cell face. By physical we mean those faces or cells that lie within the actual spatial domain, i.e. no phantoms. Most of our computational effort and time is spent in this step solving Riemann problems exactly. Step 4 handles rarefactions where they occur. Once the solution is obtained along each cell face, we calculate the Godunov flux in step 5 using Equation (66).

Finally, we calculate \bar{U}_i^{n+1} in each cell using Equation (59). Next, we will look at steps 1, 2, 4 and 6 in greater detail. Steps 3 and 5 are straightforward.

Step 1: Phantom Cells and Boundary Conditions

Until now, we have ignored the numerical difficulties we face along the boundaries of our spatial domain. Namely, border cells, 1 and I , cannot be updated using Equation (59) without \bar{F}_0 and \bar{F}_L . To define these fluxes, we apply boundary conditions along $x = 0m$ and $x = L$.

Traditionally, these boundary conditions manifest themselves in the form of boundary functions that explicitly define the unknown quantities of interest. Here, we take a different approach and define cell-averaged states

$$\bar{W}_0^n = \begin{pmatrix} \rho_1^n \\ \pm u_1^n \\ p_1^n \end{pmatrix} \quad (67)$$

and

$$\bar{W}_{I+1}^n = \begin{pmatrix} \rho_I^n \\ \pm u_I^n \\ p_I^n \end{pmatrix} \quad (68)$$

within phantom cells as depicted in Figure 16 (Toro, 1997).

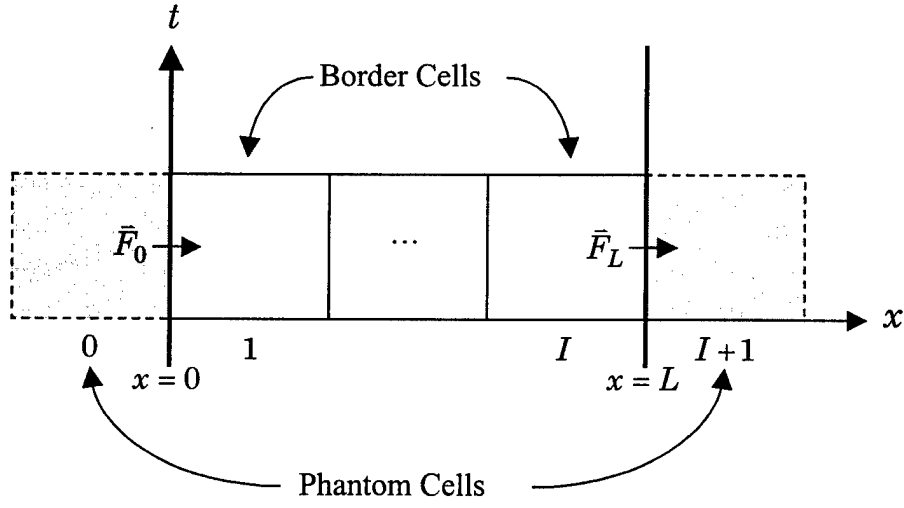


Figure 16: Phantom Cells

In modeling shocks, we consider only two types of boundaries: physical boundaries that are reflective and computational boundaries that are transmissive. To model the reflection of shocks and other waves at a perfectly reflective physical barrier, we set the primitive variable states within each phantom cell in the following manner:

$$\bar{W}_0^n = \begin{pmatrix} \rho_1 \\ -u_1 \\ p_1 \end{pmatrix} \text{ for a reflective boundary along } x = 0 \quad (69)$$

and

$$\bar{W}_{I+1}^n = \begin{pmatrix} \rho_I \\ -u_I \\ p_I \end{pmatrix} \text{ for a reflective boundary along } x = L. \quad (70)$$

The solution at reflective boundaries will consist of either two-shocks or two-rarefactions.

Transmissive boundaries are applied to restrict our physical problem to a finite computational domain. Ideally, the flows should propagate through hypothetical boundaries without any effect. For transmissive boundaries we simply set the unknown phantom state equal to that in the corresponding border cell, i.e.

$$\bar{W}_0^n = \bar{W}_1^n \text{ for a transmissive boundary at } x = 0m \quad (71)$$

or

$$\bar{W}_{I+1}^n = \bar{W}_I^n \text{ for a transmissive boundary at } x = L. \quad (72)$$

This technique works reasonably well in one-dimension. In multiple dimensions however, noticeable computational, or *false*, reflections occur. The inability to model transmissive BCs in multiple dimensions has led to a significant and ongoing research effort.

With the state of the phantom cells determined, local Riemann problems may now be solved along the boundaries to calculate the unknown fluxes, \bar{F}_0 and \bar{F}_L .

Step 2: Determination of a Stable Time Step

Before we can move forward in time to solve Riemann problems and update the solution an appropriate time step must be determined that meets

the Courant-Friedrichs-Lewy stability requirement (Courant and Friedrichs, 1948):

$$S_{\max}^n \frac{\Delta t}{\Delta x} \leq 1 \quad (73)$$

where S_{\max}^n is the maximum wave speed that occurs across the computational spatial domain. The importance of this condition in properly selecting Δt becomes evident if we consider adjacent RPs. In Figure 17, the solution to the left Riemann problem has a strong shock wave traveling rapidly to the right. If Δt is too large as depicted in Figure 17, the shock wave travels far enough to the right that it perturbs the solution we seek along face $i + 3/2$. This violates the assertion made earlier that

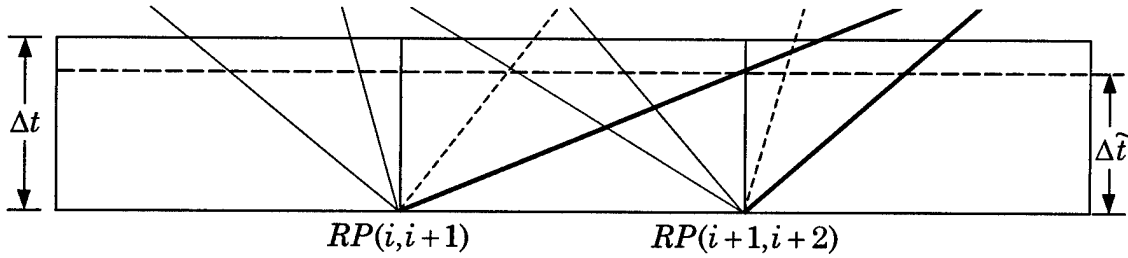


Figure 17: Wave Propagation Through Adjacent RPs

$\bar{W}(x_{i+1/2})$ is constant. This will result in an incorrect flux calculation creating numerical errors that propagate outward affecting other cells. As a result, the code rapidly becomes unstable and crashes. To prevent this, the time step is reduced to $\Delta \tilde{t}$ as shown.

Ideally, we would like to maximize the computational efficiency of the code by taking the largest time step possible:

$$\Delta t = \frac{\Delta x}{S_{\max}^n} \quad (74)$$

where S_{\max}^n satisfies Equation (73) exactly. Unfortunately, the maximum wave speed is not known a priori but must be estimated (Toro, 1997):

$$\tilde{S}_{\max}^n = \max_i (|u_i| + c_i). \quad (75)$$

This estimate is chosen because it extends easily into multiple dimensions. However, it requires some caution.

During the first few time steps, our estimate is dominated by the local speed of sound and results in a significant underestimate of S_{\max}^n . If we simply substitute \tilde{S}_{\max}^n into Equation (74) we will calculate a Δt that is too large and the code will become unstable. To prevent such an occurrence, we introduce a scaling factor, C , called the Courant coefficient into Equation (74) that satisfies $0 < C \leq 1$. Each time step is then calculated adaptively using Equation (56). During the first several time steps, Toro suggests $C = 0.2$ (1997). At later time steps, the Courant coefficient is set to 0.9 to increase Δt and computational efficiency while maintaining stability.

Step 4: Sonic Rarefactions – A Special Case

When solving RPs, we must consider a special case - sonic rarefactions. These rarefactions occur when the head and tail of the expansion fan travel

in opposite directions. When this occurs, the solution lies within the expansion fan. In order to calculate the solution, we must determine the directions of travel and apply the appropriate rarefaction equation, Equation (44) or (48), presented in Chapter 3.

Step 6: Update Cell Solution

At this point, the solution to all the Riemann problems are known, the solution obtained along each cell face and the corresponding fluxes calculated. It is important to remember that while the primitive variable form is used in solving the Riemann problems and calculating the unknown fluxes, only the conservative formulation will result in an accurate solution. The solution, \bar{U}_i^{n+1} , is calculated using Equation (59) given \bar{U}_i^n and fluxes $\bar{F}_{i-1/2}$ and $\bar{F}_{i+1/2}$. Before the next time step begins, \bar{W}_i^{n+1} is calculated from \bar{U}_i^{n+1} .

F. Illustration of Godunov's Method

To illustrate the important numerical properties of Godunov's method, we revisit the classic shock tube problem presented in Chapter 3. Recall that the solution consists of a left traveling rarefaction, right traveling contact surface and a right traveling shock wave. The solution was obtained after 125 time steps, $t = 0.25s$, using a coarse computational mesh of 200 cells. During the first five time steps, the Courant coefficient was set to 0.2. After that, it was set to 0.9. Boundary conditions were transmissive.

We present the density profile obtained via Godunov's method in Figure 18. For comparison, the exact solution obtained in Chapter 3 is shown as well. The solution is smeared near each discontinuity. This undesirable

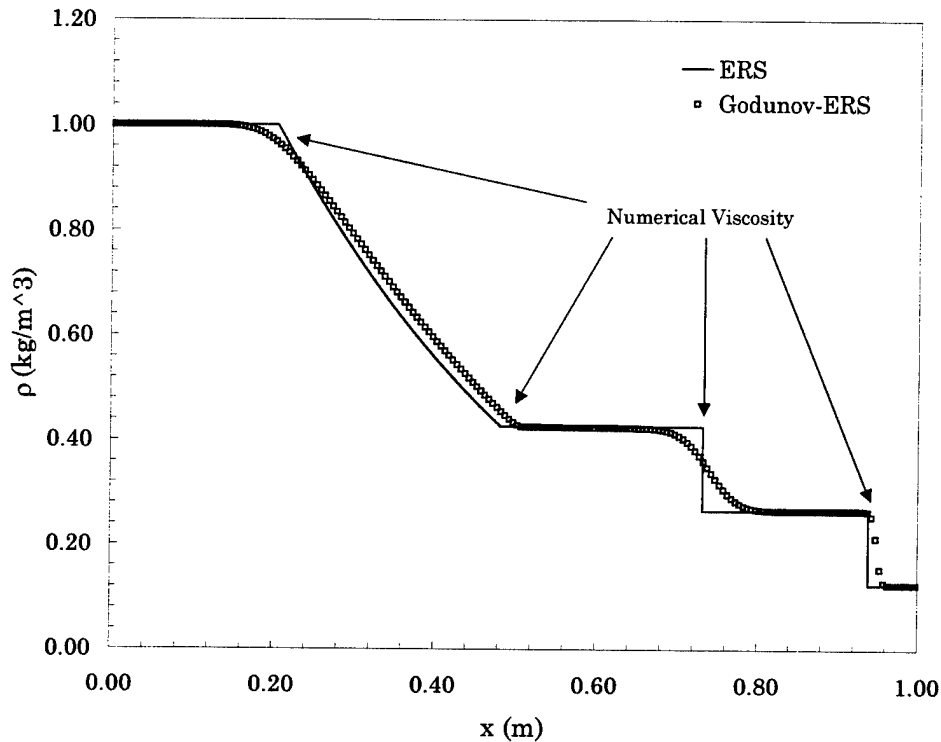


Figure 18: Godunov vs ERS - Density Profile at $t=0.25s$

behavior is common among first-order methods and is caused by excessive numerical viscosity.

Clearly, this smearing is not equal in severity everywhere. Notice that the shock front is spread over six cells while the contact surface occurs over 23 cells. This disparity in resolution is common among Riemann problem type methods and is a direct result of the characteristic nature of each wave. For a contact wave, the characteristics run parallel to one another and it is

the difference in internal energy that results in the density discontinuity. Therefore, the contact discontinuity is not easily resolved. Shock wave resolution is better because the characteristics run into each other. The smearing that occurs along the rarefaction fan occurs at the head and tail where the first derivative of the solution is discontinuous. The rarefaction wave becomes resolved quickly as we refine the mesh.

Before we leave this section, several positive features of Godunov's method are pointed out. The position and speed of each wave are predicted accurately. This is very important because we want to model shocks correctly. The monotonic behavior of the solution near discontinuities is another important quality. This feature becomes important later when we apply higher-order techniques. Lastly, while numerical viscosity is evident, most first-order methods result in significantly more smearing than we observe here.

G. Improvements in Computational Efficiency and Accuracy

Adaptive Riemann Solver (ARS)

In Section F, we found Godunov's method to be first-order accurate. Given the large computational effort spent solving each Riemann problem exactly, this method seems dubious at best. The obvious question comes to mind. Why put forth all that effort to achieve only first-order accuracy? Fortunately, over the last two decades, several algorithms have been developed that approximate the solution to the Riemann problem with

accuracy and efficiency. These approximate Riemann solvers have given rise to an entire class of techniques called Godunov-type methods. Approximate Riemann solvers are used in one of two ways:

- they are used to approximate the numerical flux directly
- they are used to approximate the state, $\bar{W}(x_{i+1/2})$.

Here, we use an adaptive scheme that switches between three approximate Riemann solvers to find a state approximation. We then calculate the Godunov flux and proceed as described in Section E.

The adaptive Riemann solver scheme was first introduced by Toro in 1997. We have implemented this particular algorithm because it

- is easily implemented within our code – two of the approximate Riemann solvers are derived directly from the exact Riemann solver,
- is very efficient computationally – none of the approximate Riemann solvers require iteration,
- captures shocks well – does not require shock fitting or entropy fix common among other techniques and it
- predicts the appropriate solver for each RP adaptively using a technique we have already employed and found successful (see appendix A).

As stated above, ARS selects one of three approximate Riemann solvers based upon the local flow conditions present and the behavior of Toro's pressure function. In fact, ARS uses the same switching criteria introduced in appendix A to predict an initial guess for the Newton-Raphson root solver. We summarize that logic here:

1. Calculate p_{\min} , p_{\max} and p_{purs} .
2. If $p_{\min} \leq p_{purs} \leq p_{\max}$ and $Q_S \geq p_{\max}/p_{\min}$ then p_* lies within I_2 and mild conditions exist. Approximate R_* using the primitive variable solver.
3. If however, $p_{purs} < p_{\min}$ then p_* lies within I_1 and two rarefactions are present. Approximate R_* using the two-rarefaction solver.
4. Else, p_* lies within I_3 and either two shocks or a single strong shock are present. Approximate R_* using the two-shock solver.

The three approximate Riemann solvers of the adaptive Riemann solver are:

- the primitive variable Riemann solver,
- the two-rarefaction Riemann solver and
- the two-shock Riemann solver.

We present the solution method below for each.

Primitive Variable Riemann Solver (PVRs)

In typical shock problems, most of the flow field varies smoothly. If the solution lies within I_2 and no strong shocks are present, we may

approximate the solution by solving a linear hyperbolic system of PDEs exactly. Toro's primitive variable solver is derived in this fashion and results in a set of simple algebraic expressions (1991). Using PVRS, we approximate the solution in R_* using the following relations (Toro, 1991):

$$p_* = \frac{1}{2}(p_1 + p_4) + \frac{1}{8}(u_1 - u_4)(\rho_1 + \rho_4)(c_1 + c_4), \quad (76)$$

$$u_* = \frac{1}{2}(u_1 + u_4) + \frac{2(p_1 - p_4)}{(\rho_1 + \rho_4)(c_1 + c_4)}, \quad (77)$$

$$\rho_2 = \rho_1 + \frac{(u_1 - u_*)(\rho_1 + \rho_4)}{(c_1 + c_4)} \quad (78)$$

and

$$\rho_3 = \rho_4 + \frac{(u_* - u_4)(\rho_1 + \rho_4)}{(c_1 + c_4)}. \quad (79)$$

Two-Rarefaction Riemann Solver (TRRS)

If two rarefactions are present within the solution, we can find p_* exactly using the appropriate rarefaction relations in Equation (30). Using the two-rarefaction solution, we approximate the solution within R_* as (Toro, 1997):

$$p_* = \left(\frac{c_1 + c_4 - \frac{1}{2}(\gamma - 1)(u_4 - u_1)}{\frac{c_1}{p_1^{(\gamma-1)/2\gamma}} + \frac{c_4}{p_4^{(\gamma-1)/2\gamma}}} \right)^{1/2}, \quad (80)$$

$$u_* = \frac{1}{2}(u_1 + u_4) + \frac{1}{2}(f_4 - f_1) \quad (81)$$

and

$$d_{\xi \pm 1}(p_*, \rho_\xi, p_\xi) = \begin{cases} \rho_\xi \left(\frac{p_*(\gamma + 1) + p_\xi(\gamma - 1)}{p_*(\gamma - 1) + p_\xi(\gamma + 1)} \right) & \text{if } p_* > p_\xi \quad \text{shock} \\ \rho_\xi \left(\frac{p_*}{p_\xi} \right)^{1/\gamma} & \text{if } p_* \leq p_\xi \quad \text{rarefaction} \end{cases} \quad (82)$$

Two-Shock Riemann Solver (TSRS)

If two shocks or a single strong shock are present, we use the two-shock approximation. Direct substitution of the appropriate shock relations into Equation (30) results in the two-shock approximation. Unfortunately, this approximation is dependent upon p_* itself and therefore, does not have a closed-form solution. An acceptable solution however, is to estimate p_* first using the primitive variable result described above. This requires no additional effort since we have already performed this estimate in determining which solver to use. We simply substitute this estimate into our two-shock approximation. We now have an adequate approximation without iteration.

We approximate the R_* variables using the two-shock solver as (Toro, 1997):

$$\begin{aligned}
p_{\star}^{(0)} = & \left[p_1 \left(\frac{2}{\rho_1(p_{\star}^{(0)}(\gamma+1) + p_1(\gamma-1))} \right)^{1/2} + \right. \\
& \left. p_4 \left(\frac{2}{\rho_4(p_{\star}^{(0)}(\gamma+1) + p_4(\gamma-1))} \right)^{1/2} + u_1 - u_4 \right] \times \\
& \left[\left(\frac{2}{\rho_1(p_{\star}^{(0)}(\gamma+1) + p_1(\gamma-1))} \right)^{1/2} + \left(\frac{2}{\rho_4(p_{\star}^{(0)}(\gamma+1) + p_4(\gamma-1))} \right)^{1/2} \right]^{-1}
\end{aligned} \tag{83}$$

where

$$p_{\star}^{(0)} = \max(1 \times 10^{-6} Pa, p_{pvrs}) \tag{84}$$

To obtain the most accurate solution, we calculate the remaining star variables as described above in TRRS.

Adaptive Riemann Solver Results

We return to Sod's problem and solve it again, using the initial conditions and choice of C . In Figure 19, we compare the density profile obtained using Godunov's method with both the exact and adaptive solvers. Graphically, we can not differentiate between the two. Each results in the same first-order behavior described in Section F.

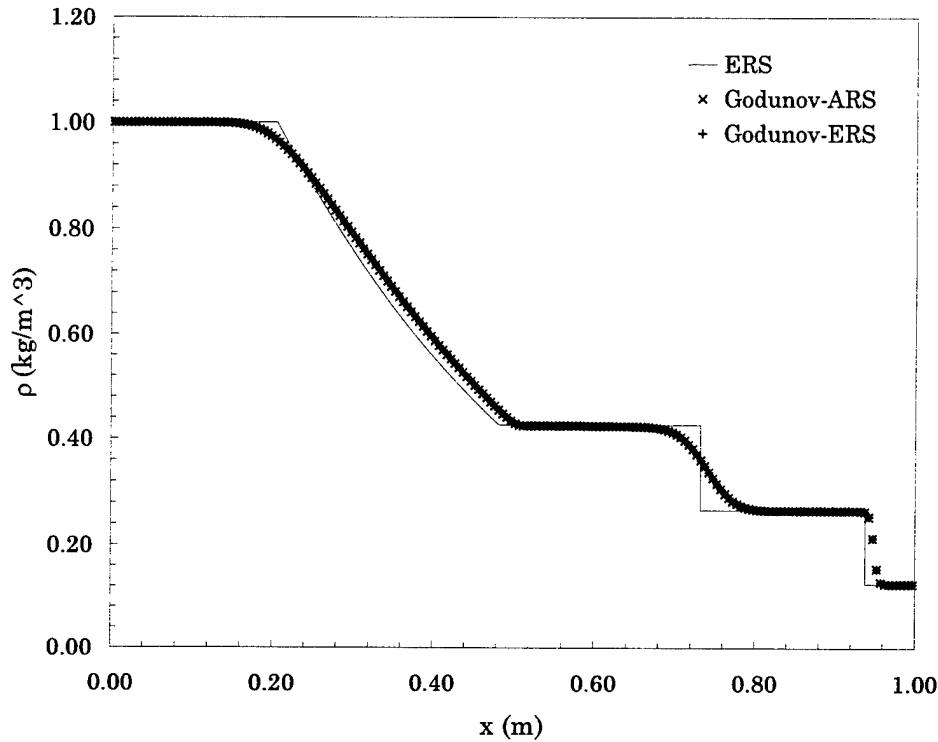


Figure 19: Comparison of Godunov's Method Using ERS and ARS

Computationally however, the difference is rather significant.

Choosing the adaptive routine results in a 33% increase in computational efficiency as measured by the amount of CPU time spent in solving the problem, shown in Table 2. Note that many of the Riemann problems are solved using the primitive variable method. This is the simplest of the three approximate solvers used in the adaptive scheme. Additional test cases and problems have been solved using both solvers and the results observed are consistent in every case. From these results, we conclude the adaptive solver outperforms the exact one and will be used as the Riemann solver.

Method	CPU Time (s)	ERS %	PVRS %	TRRS %	TSRS %
ERS	1.28	100	0	0	0
ARS	0.859	0	90	5	5

Table 2: Comparison Between the ERS and ARS

Adaptive Courant Coefficient

In an effort to improve computational efficiency and accuracy, we briefly investigate an adaptive routine that sets the Courant coefficient, C , adaptively. The original goal of this effort was to maximize the time step from the very beginning while maintaining numerical stability over the entire temporal domain. Traditionally, C is set to a small positive number, often $C = 0.2$, to correct the underestimate of S_{\max}^n . After several time steps, usually 5, C is set to 0.9 to maximize our time steps while maintaining stability. Our concern in using the simplistic approach above is twofold:

- why waste computational efficiency in the early stages of code execution – can we increase C sooner, say after one time step?
- once we set C to 0.9 or some other value determined adaptively, can complex flows ever result in a loss of stability?

Here, we investigate two possible adaptive techniques to maximize the time step while maintaining stability. At the beginning of each time step, we assume the value of C scales the right hand side exactly resulting in the

maximum allowable time step. Once all the Riemann problems are solved, we calculate the maximum local Courant number,

$$\nu_{\max}^n = \max_i \left(\frac{S_i^n \Delta t}{\Delta x} \right). \quad (85)$$

If $\nu_{\max}^n \leq 1$, the time step satisfies Equation (73), we reset C and continue our calculations. It is the method chosen in resetting C that defines the different adaptive methods. In the naive approach, we assume flows are sufficiently established over the current time step, t^n to t^{n+1} . This implies that, at t^{n+1} , \tilde{S}_{\max}^{n+1} is a reasonable estimate. In this case, we simply set $C^{n+1} = C_{\max}$ where C_{\max} is the user-defined maximum allowable Courant coefficient.

A more sophisticated method assumes flow conditions will remain essentially the same over the next time step. This suggests $S_{\max}^n \cong S_{\max}^{n+1}$ and allows us to use the currently obtained solution to scale C in the following manner:

$$C^{n+1} = \min \left(\frac{C_{\max} - \nu_{\max}^n}{2} + \nu_{\max}^n, C_{\max} \right). \quad (86)$$

We choose the minimum to handle the special case where $C_{\max} < \nu_{\max}^n \leq 1$.

Given typical flow conditions, we expect C will rapidly approach C_{\max} asymptotically.

If $v_{\max}^n > 1$ however, our assumptions concerning C were inappropriate. In this case we throw away the solution, reduce C^n and begin again. In each adaptive routine described above, we reset C in the following manner:

$$C^n = \left(\frac{C^n}{v_{\max}^n} \right) C_{\max}. \quad (87)$$

Adaptive Courant Results

Here, we solve Sod's problem via Godunov's method (with ARS) using the typical method as suggested by Toro, the naïve adaptive Courant method and the sophisticated adaptive Courant method. We discretized the spatial domain over 800 cells to increase the number of time steps.

Method	CPU Time	Time Steps	Restarts
Toro	5.2075	490	0
naïve	5.218	486	1
sophisticated	5.238	487	0

Table 3: Adaptive Courant Results - Sod Test

We see in Table 3 that Toro's and our naïve method 3 calculate stable time steps throughout the temporal domain. The naïve approach however, required a restart. This occurred at the second time step because flows were not yet established sufficiently to predict S_{\max}^2 resulting in an unstable time step during our first attempt. CPU times increase as we increase the complexity of the time step calculations. When compared to the number of

time steps saved, our results indicate that adaptive time step methods are counterproductive.

To investigate the behavior of C in the presence of complicated flows, we solve the classic Woodward-Colella blast problem that consists of two strong shocks traveling in air towards each other (Woodward and Colella, 1984). These shocks interact with each other and reflect from reflective boundaries resulting in stagnation, reflection and transmission of shock waves. The problem begins at $t = 0s$ with three initial states at rest. These states are separated by two discontinuities located at $x = 0.1m$ and $x = 0.9m$. The physical domain is $1m$ in length and lies between two reflective barriers. The domain is discretized over 1000 cells. The following initial conditions are:

$$\bar{W}(x,0) = \begin{cases} \bar{W} = (1.00 kg/m^3, 0.00 m/s, 1000.00 Pa)^T & 0.0m < x < 0.1m \\ \bar{W} = (1.00 kg/m^3, 0.00 m/s, 0.01 Pa)^T & 0.1m < x < 0.9m \\ \bar{W} = (1.00 kg/m^3, 0.00 m/s, 100.00 Pa)^T & 0.9m < x < 1.0m \end{cases} \quad (88)$$

After 0.05s, the shocks have reflected from the reflective boundaries and we terminate program execution. Table 4 shows our results for the same three methods we considered above. Our results are similar with those obtained in Table 3 despite the presence of complex flows; CPU time increases as each case increases in complexity. Notice that the naïve method requires no restart. The absence of a restart is explained when we consider the large

pressure gradients involved at early times. This results in a satisfactory \tilde{S}_{\max}^n after only one time step. Our results suggest there is no need for an adaptive Courant coefficient in the presence of mild or strong flows. In fact, using an adaptive method decreases the computational efficiency of the code therefore we omit it.

Method	CPU Time	Time Steps	Restarts
Toro	40.59	2155	0
naïve	40.81	2151	0
sophisticated	42.51	2152	0

Table 4: Adaptive Courant Results - Sod Test

H. Toro's Weighted Average Flux (WAF) Method

Now that we have implemented an efficient form of Godunov's method, we present Toro's second-order extension called WAF. While WAF can be applied equally to many first-order techniques, we apply it here with Godunov's method to achieve second-order accuracy. Its implementation is numerically simple and straightforward.

Recall that in Godunov's method we calculate the numerical flux using the solution of the Riemann problem obtained along the cell face only. Much of the characteristic information obtained in solving the Riemann problem has not been utilized, yet. WAF however, approximates the total flux through each cell face using a spatial quadrature scheme across the entire wave structure of the Riemann problem. Using all the characteristic

information allows us to calculate a numerical flux that is more accurate.

Empirical results show this scheme is second-order accurate.

We calculate WAF in the following manner (Toro, 1989):

$$\bar{F}_{i+1/2} = \sum_k w_k \bar{F}_k \quad (89)$$

where w_k is a numerical weight that represents the geometric extent of region k and

$$\bar{F}_k = \bar{F}(\bar{W}_k) \quad (90)$$

is the flux through region k . Figure 20 illustrates WAF in the $x-t$ plane for the Riemann problem at face $i+1/2$.

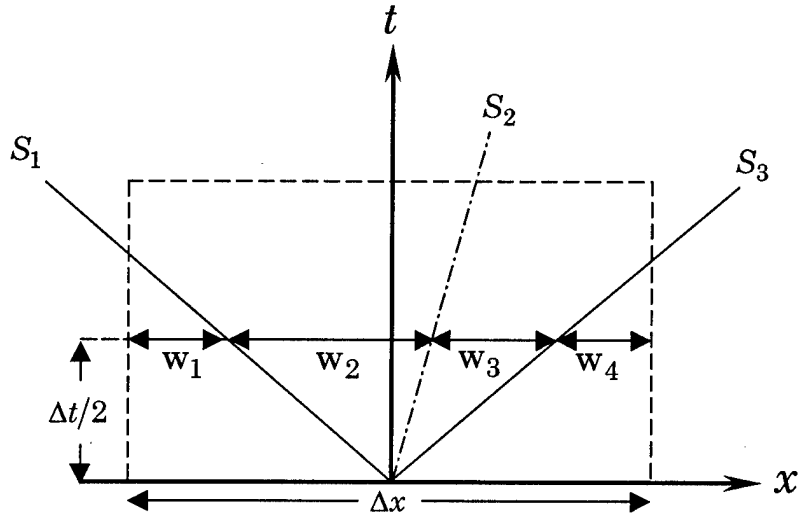


Figure 20: Weights of WAF

At $t^{n+1/2}$, we see the Riemann problem spatial domain is subdivided into four subintervals of length $w_k \Delta x$. Notice that we depict the outer waves with only one ray instead of two. Empirical evidence has shown that the

expansion fan solution contributes very little to the overall accuracy of the solution. Therefore, we can ignore the expansion fan collapsing it into the region nearest the cell face, i.e. R_2 or R_3 . This results in the simplified wave structure shown in Figure 20. Wave speeds, S_1 and S_3 , remain the same as before. The weights are defined as follows (Toro, 1998):

$$w_1 = \frac{1}{2}(1 + v_1), \quad (91)$$

$$w_2 = \frac{1}{2}(\nu_2 - \nu_1), \quad (92)$$

$$w_3 = \frac{1}{2}(\nu_3 - \nu_2) \quad (93)$$

and

$$w_4 = \frac{1}{2}(1 - \nu_3) \quad (94)$$

where

$$\nu_k = \frac{S_k \Delta t}{\Delta x} \quad (95)$$

is local Courant number for the k^{th} wave. Note that $\sum_k w_k = 1$ and that the role each partial flux plays in determining $\bar{F}_{i+1/2}^{WAF}$ is dependent upon ν_k . In practice, we implement WAF in the following manner:

1. Calculate weighted average state along face $i + 1/2$ where

$$\bar{W}_{i+1/2}^{WAS} = \sum_k w_k \bar{W}_k$$

2. Calculate weighted average flux from our weighted average state,

$$\text{i.e., } \bar{F}_{i+1/2}^{WAF} = \bar{F}(\bar{W}_{i+1/2}^{WAS}).$$

This form of WAF is much easier to implement.

I. Illustration of WAF

To illustrate WAF we solve Sod's problem and present the density profile again. In Figure 21, we see improved resolution of discontinuities over that achieved using Godunov's method alone. It is immediately obvious however, that oscillation occurs near each discontinuity. This behavior is quite typical of second-order techniques. To explain the origin of these oscillations, we return to the definition of WAF.

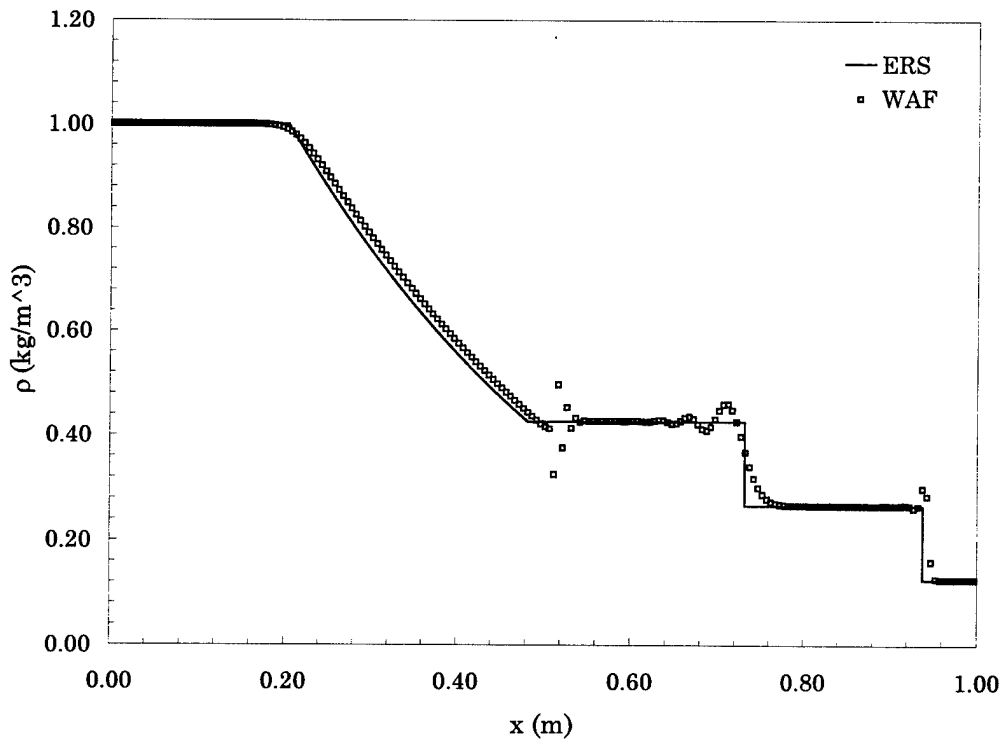


Figure 21: WAF - Density Profile at t=0.25s

Close inspection of Equation (89) provides some insight into WAF and the oscillatory nature of the solution. Let's consider the Riemann problem in Figure 20. Notice that the local t axis lies within the second region and therefore, $\bar{F}_2 = \bar{F}^{GF}$. This illustrates an important fact; Godunov's flux is always represented by one of the fluxes in Equation (89). Note that it is the local Riemann problem structure that determines which flux corresponds to Godunov's flux.

\bar{F}^{GF} represents the upwind bias of WAF and is responsible for stability. The other fluxes represent downwind terms and are responsible for the increased accuracy we observe with WAF. It is these downwind terms that cause the oscillations we observe near steep gradients. In the next section, we present an adaptive technique that controls the contribution of these downwind terms near discontinuities resulting in near monotonic conditions.

J. Limited-WAF (LWAF)

Here, we present total variation diminishing (TVD) weight limiter functions that restrict the downwind partial flux contributions near discontinuities by introducing dissipative viscosity (Toro, 1997). This results in near monotonic conditions throughout the domain.

Before we continue, we briefly define what TVD means. A numerical scheme is considered to be TVD in nature when the total variation of the solution diminishes over time (LeVeque, 1992). In mathematical terms,

$$TV(\bar{U}^{n+1}) \leq TV(\bar{U}^n) \quad (96)$$

where the total variation function, TV , of the solution at t^n is given by:

$$TV(\bar{U}^n) = \sum_i \left| \bar{U}_{i+1}^n - \bar{U}_i^n \right|. \quad (97)$$

TVD numerical schemes have several important features:

- they are TV stable meaning they are guaranteed to converge to the appropriate solution.
- the numerical scheme decreases toward a monotonic solution.

Monotonic-like second-order numerical schemes are derived from these principles.

To achieve monotonic-like conditions near discontinuities, Toro has developed TVD weight limiter functions, $\phi(r, \nu)$, that are based upon TVD flux limiter functions, $\psi(r)$. Any weight limiter may be derived given the corresponding flux limiter using the following relation (Toro, 1997):

$$\phi(r, \nu) = 1 - (1 - |\nu|) \psi(r) \sigma(\nu) \quad (98)$$

where r is a local flow parameter we will discuss later and $\sigma = \text{SGN}(\nu)$.

Here, we construct the following four weight limiter functions: Super A (ϕ_{SA}), van Leer A (ϕ_{VLA}), van Albada A (ϕ_{VAA}) and Min A (ϕ_{MA}). Each is

constructed using Equation (98) from the corresponding flux limiter functions (Toro, 1997):

$$\text{Superbee: } \psi_{SB} = \begin{cases} 0 & r \leq 0 \\ 2r & 0 \leq r \leq 0.5 \\ 1 & 0.5 \leq r \leq 1, \\ r & 1 \leq r \leq 2 \\ 2 & r \geq 2 \end{cases} \quad (99)$$

$$\text{van Leer: } \psi_{VL} = \begin{cases} 0 & r \leq 0 \\ \frac{2r}{1+r} & r \geq 0 \end{cases}, \quad (100)$$

$$\text{van Alabada: } \psi_{VA} = \begin{cases} 0 & r \leq 0 \\ \frac{r(1+r)}{1+r^2} & r \geq 0 \end{cases}, \quad (101)$$

and

$$\text{Minbee: } \psi_{MB} = \begin{cases} 0 & r \leq 0 \\ r & 0 \leq r \leq 1. \\ 1 & r \geq 1 \end{cases} \quad (102)$$

Before we describe the weight limiters, it is helpful to investigate how the flux limiters work.

Each flux limiter is calculated based upon the local flow parameter, r , defined as (Sweby, 1984):

$$r \equiv \frac{\Delta \rho_{upwind}}{\Delta \rho_{local}} \quad (103)$$

where $\Delta\rho_{local}$ is the density change from right to left across a specific wave in the Riemann problem of interest and $\Delta\rho_{upwind}$ is the change in density across the same wave in the adjacent upwind Riemann problem. In WAF, we calculate local flow parameters for each wave in the Riemann problem at face $i+1/2$ such that

$$r_{i+1/2,k} = \frac{\Delta\rho_{i+1/2-\sigma,k}}{\Delta\rho_{i+1/2,k}} \quad (104)$$

where the local density change across each wave is given by:

$\Delta\rho_{i+1/2,k} = \rho_{i+1/2,k+1} - \rho_{i+1/2,k}$. Figure 22 illustrates how $r_{i+1/2,k}$ is calculated.

In case 1,

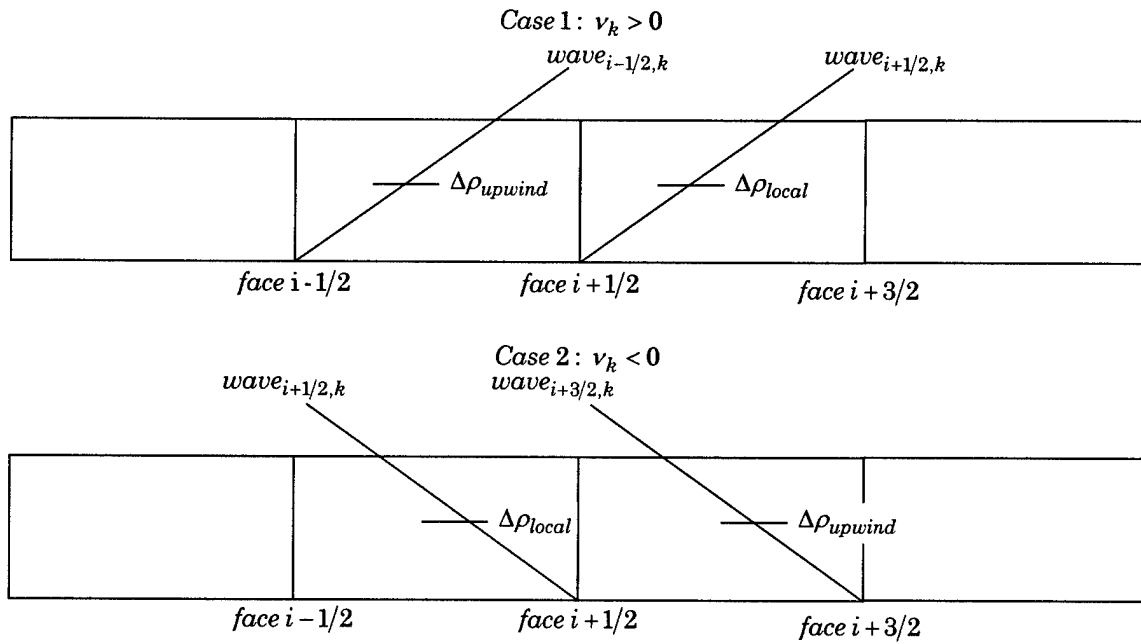


Figure 22: Calculation of Local Flow Parameters

$v_k > 0$ meaning k^{th} wave's are traveling to the right. In this case, $\sigma = 1$

identifying $wave_{i-1/2,k}$ as the upwind wave and $r_{i+1/2,k} = \frac{\Delta\rho_{i-1/2,k}}{\Delta\rho_{i+1/2,k}}$. In case 2

however, $v_k < 0$ and $\sigma = -1$ so that $wave_{i+3/2,k}$ is the upwind wave and

$r_{i+1/2,k} = \frac{\Delta\rho_{i+3/2,k}}{\Delta\rho_{i+1/2,k}}$. In each case, r is a measure of how rapidly the density

gradient changes locally. This is a measure of the smoothness of the flow, i.e.

as $r \rightarrow 1$, the flow becomes smooth. In these regions we include the

downwind flux contributions to achieve second-order accuracy. As

$|1 - r_k| \gg 0$, the flow becomes discontinuous. In these regions we severely

limit the contribution of all second-order downwind terms to $\bar{F}_{i+1/2}^{WAF}$ to

achieve stability and monotonic behavior.

Weight limiters work in a similar manner. Instead of limiting the flux contributions directly however, weight limiters amplify courant numbers, v_k , depending upon local flow conditions as follows:

$$\phi_k = \bar{v}_k = \alpha_k v_k \quad (105)$$

where $\alpha_k = \frac{1 - (1 - |v_k|) \psi(r_k)}{|v_k|}$ is the TVD wave amplifier for the k^{th} wave

(Toro, 1992). Amplification of these Courant numbers changes the value of

the corresponding weights. For example, as $|1 - r_k|$ becomes large, the

corresponding weight approaches zero effectively killing off the downwind contribution of that flux.

Boundary conditions in the limited WAF scheme must be treated with care. In order to account for the upwind calculations required by the limited WAF scheme, phantom cells -1 and $I+2$ must be added. For example, consider the calculation of the limited flux at $x = 0m$. If $v_{1/2,k} > 0$, then we must add cell -1 to solve the Riemann problem located at $x = -\Delta x$.

K. Illustration of Limited WAF

Now that we have developed the limited version of WAF, we solve Sod's problem using each weight limiter function to briefly investigate the numerical properties of each. The results are shown in Figures 23 – 25. In Figure 23 we see that every limiter dampens out the second-order partial flux terms near discontinuities. It is however, difficult to pick out any details concerning the performance of each limiter near areas of interest: contact discontinuities and shocks. Figures 24 and 25 amplify these regions. It becomes evident from Figure 24 that Super A performs best near contact surfaces. In Figure 25 the behavior of each appears to be similar until we observe the behavior of each to the left of the shock. In this region we see that van Leer A performs best. The other limiter functions oscillate slightly. From these results we conclude that van Leer A is appropriate when we model shocks.

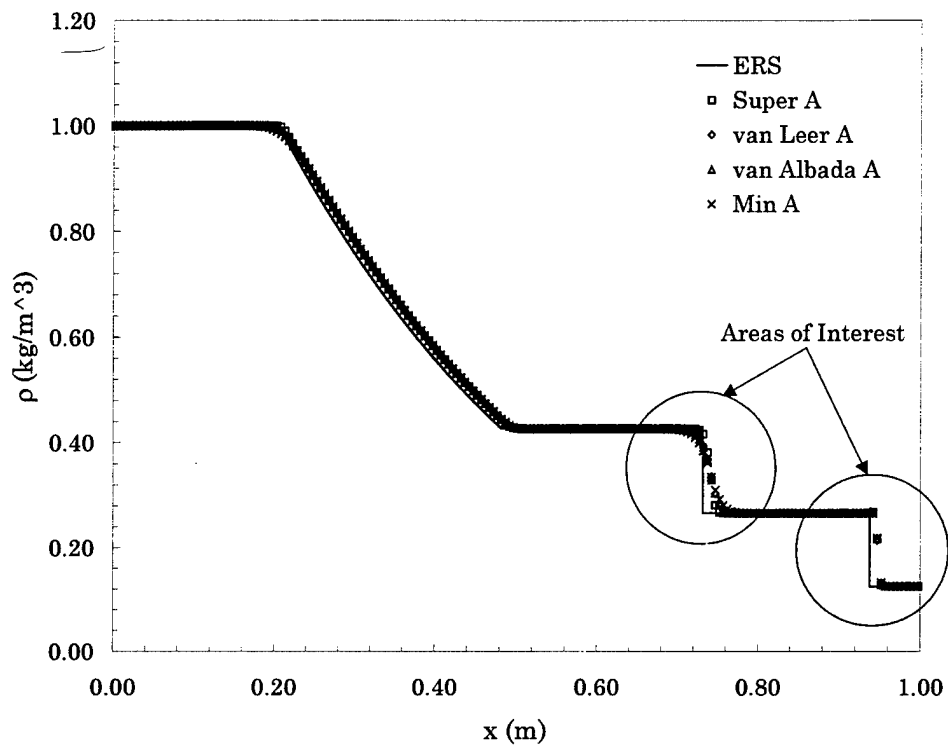


Figure 23: LWF - Density Profile at $t=0.25s$

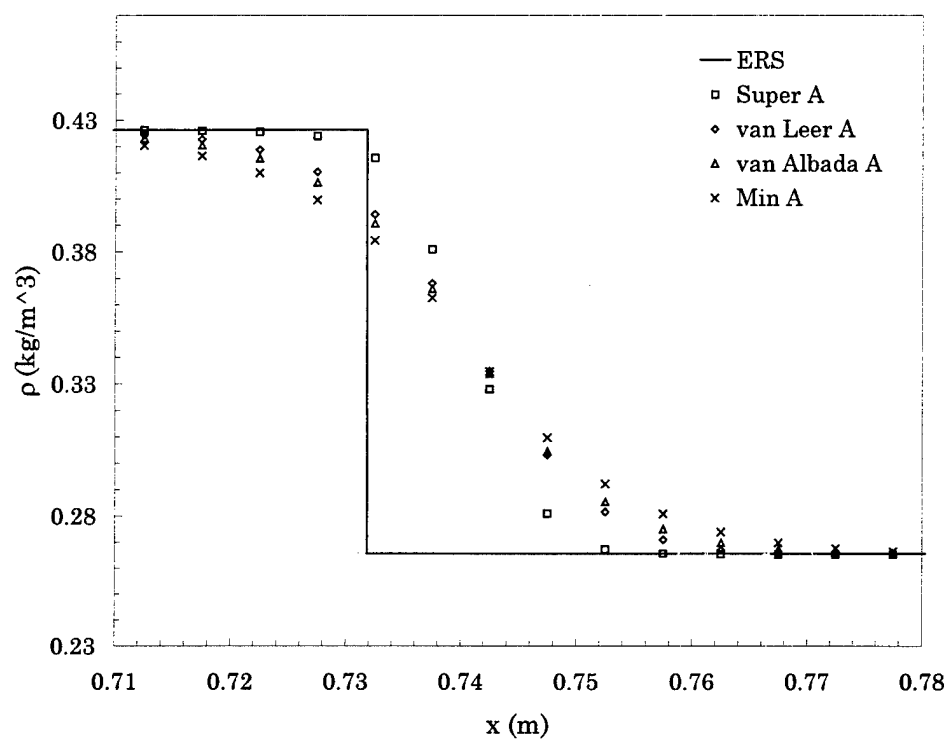


Figure 24: Comparison of Limiters near Contact Surface

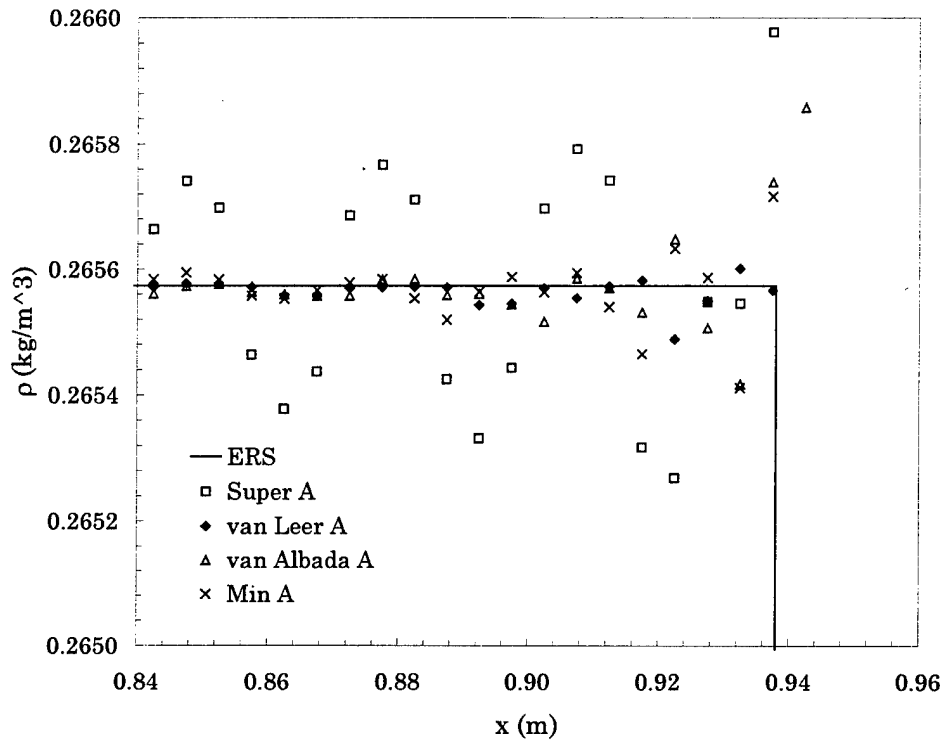


Figure 25: Comparison of Limiters near Shock

L. Testing and Validation of the One-Dimensional Shock Code

Testing

To test the implementation of the one-dimensional code we solved the five shock tube problems presented in Chapter 3 in addition to Sod's as presented here. At each step in the development of the code, we qualitatively compared our results with those obtained using the exact Riemann solver. This was done to ensure proper implementation of each numerical tool by testing it over a broad range of problems before adding additional complexity. Having tested for proper implementation, we then validated the code against experimental data.

Validation

In 1996, the Army Research Laboratory used measurements obtained in their 57cm shock tube facility to validate their own in-house second-order accurate one-dimensional shock code. Here, we use the same data to validate our own code before moving into two dimensions. The shock tube was 100m in length, 57cm in diameter and contained a driver region 0.91m long (Schraml,1996). A measurement station was located 31.48m from the initial discontinuity where overpressure time histories were taken experimentally for validation of the Army Research Laboratory code. The following conditions were present at the beginning of the experiment:

$$\bar{W}(x,0) = \begin{cases} \bar{W}_1 = (4.486 \text{ kg/m}^3, 0.0 \text{ m/s}, 379.2 \times 10^3 \text{ Pa}) & x < 0.91 \text{ m} \\ \bar{W}_4 = (1.208 \text{ kg/m}^3, 0.0 \text{ m/s}, 102.1 \times 10^3 \text{ Pa}) & x > 0.91 \text{ m} \end{cases} \quad (106)$$

Figure 26 shows the overpressure time history taken at the measurement station as compared to the Army Research Laboratory's computational results (Schraml,1996). Notice the large oscillations that occur near the shock front in Figure 26. The shock front shock arrived at the measurement station after 66.0ms. The measured peak overpressure at the shock front was 66.3kPa.

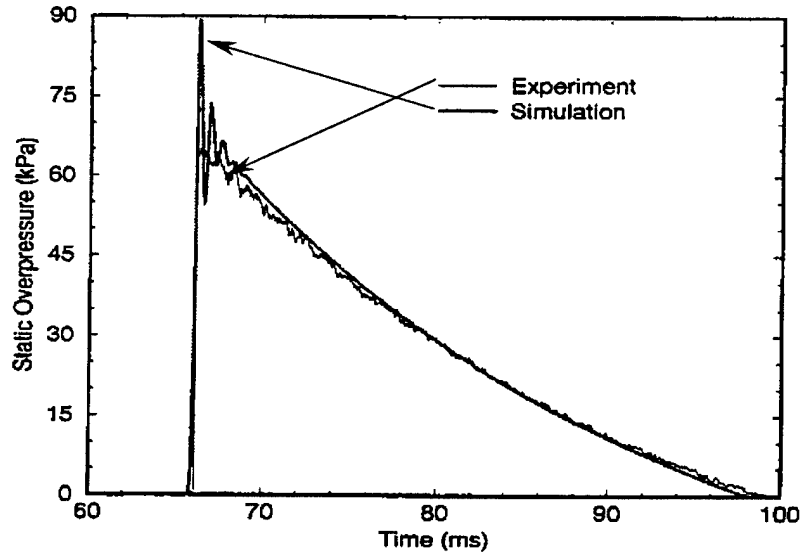


Figure 26: ARL Overpressure Time History at 31.44 m

To correctly model this experiment using our inviscid code, we had to first calibrate it to achieve the same time of arrival. We found that increasing the initial conditions within the driver region by 5% was sufficient to achieve the correct time of arrival. Therefore, the following ICs were used to model the shock tube:

$$\bar{W}(x,0) = \begin{cases} \bar{W}_1 = (4.710 \text{ kg/m}^3, 0.0 \text{ m/s}, 398.1 \times 10^3 \text{ Pa}) & x < 0.91 \text{ m} \\ \bar{W}_4 = (1.208 \text{ kg/m}^3, 0.0 \text{ m/s}, 102.1 \times 10^3 \text{ Pa}) & x > 0.91 \text{ m} \end{cases} \quad (107)$$

To model the left shock tube wall, we placed a reflective boundary at $x = 0 \text{ m}$. On the left, we restricted the computational domain by placing a transmissive boundary at $x = 50 \text{ m}$. This reduced the execution time required but was far enough from the measurement station not to affect the results.

The restricted spatial domain, $[0m, 50m]$, was discretized into 5000 cells.

C_{\max} was set to 0.9 and we used the van Leer A weight limiter function.

Figure 27 shows the overpressure time history obtained via our code. Unlike the Army Research Laboratory code results, the overpressure history curve is very smooth near the shock front. This illustrates the impact our TVD weight limiter functions.

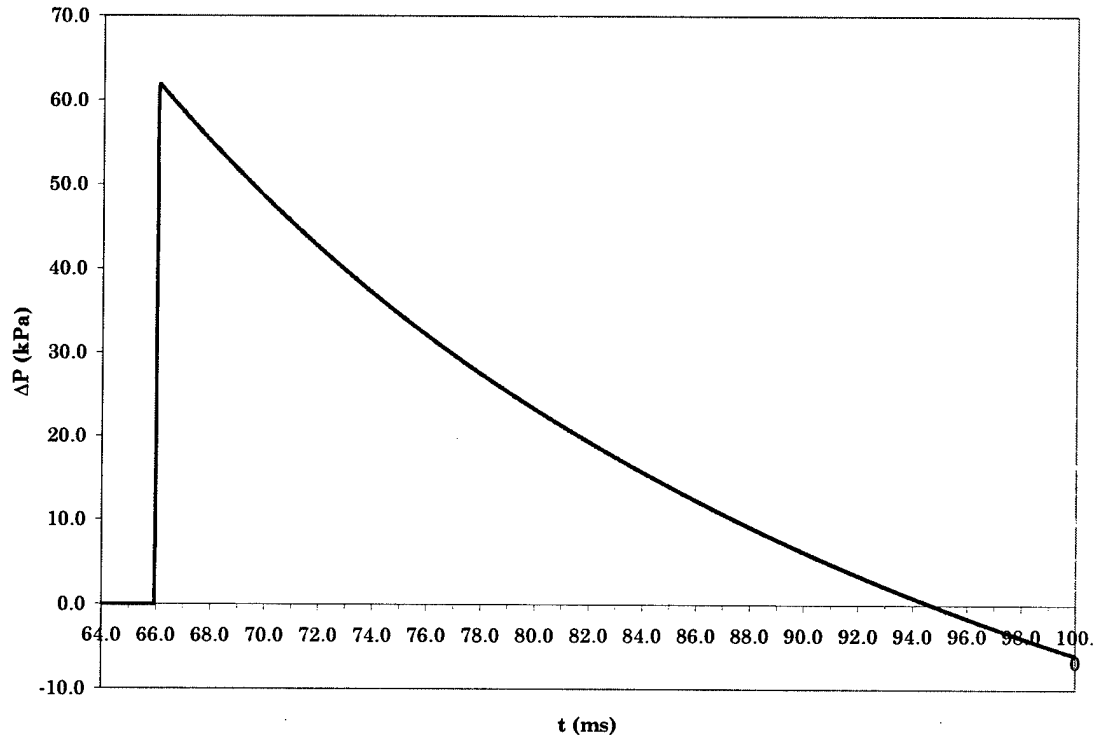


Figure 27: Overpressure Time History at 31.44m

The shock arrived at our computational measurement station, located at $x = 31.44$ m, after 66.08ms. The code predicts a peak overpressure at the shock front of 61.8kPa. This is within 6.8% of the experimental results. This is quite reasonable and suggests our code models shock propagation correctly.

Chapter 5: Shock Code Implementation in Two Dimensions

A. Initial Boundary Value Problem

In this chapter we consider and solve the two-dimensional IBVP:

$$IBVP = \begin{cases} \bar{U}_t + \bar{F}(\bar{U})_x + \bar{G}(\bar{U})_y = \bar{0} \\ \bar{U}(x, y, 0) = \bar{U}^0(x, y) \end{cases} \quad (108)$$

where $\bar{U}^0(x, y)$ is a piecewise function that defines initial conditions over the rectangular spatial domain, $[0, L_x] \times [0, L_y]$, at $t = 0$ s. In two dimensions we describe physical conditions along all four boundaries again using reflective and transmissive boundary conditions.

B. Discretization of Domain with Two Spatial Dimensions

To discretize the domain, $[0, L_x] \times [0, L_y] \times [0, T]$, we simply extend our one-dimensional mesh in the y direction. With an additional spatial direction we now have three mesh indices: i , j and n where the cell-averaged value in cell (i, j) at t^n is

$$\bar{U}_{i,j}^n \equiv \bar{U}(x_i, y_j, t^n). \quad (109)$$

The spatial domain, $[0, L_x] \times [0, L_y]$, is subdivided into $I \times J$ cells of uniform dimension, $\Delta x \times \Delta y$, where

$$\Delta x = \frac{L_x}{I} \quad (110)$$

and

$$\Delta y = \frac{L_y}{J}. \quad (111)$$

The center of cell (i, j) is located at:

$$(x_i, y_j) = \left(L_x \frac{(i-0.5)}{I}, L_y \frac{(j-0.5)}{J} \right). \quad (112)$$

The physical extent of cell (i, j) is defined by its faces. The location of each face is defined by its corresponding plane as follows:

$$\text{face } i-1/2: x_{i-1/2} = \frac{L_x(i-1)}{I}, \quad (113)$$

$$\text{face } i+1/2: x_{i+1/2} = \frac{L_x(i)}{I}, \quad (114)$$

$$\text{face } j-1/2: y_{j-1/2} = \frac{L_y(j-1)}{J}, \quad (115)$$

and

$$\text{face } j+1/2: y_{j+1/2} = \frac{L_y(j)}{J}. \quad (116)$$

The temporal domain is subdivided adaptively into N time steps of non-uniform length, Δt . Here we concern ourselves with limiting wave propagation in both spatial directions to maintain the Courant-Friedrichs-Lewy stability requirement. In two dimensions therefore, we calculate the n^{th} time step in the following manner (Toro, 1997):

$$\Delta t^n = C \min \left(\frac{\Delta x}{(\tilde{S}_i^n)_{\max}}, \frac{\Delta y}{(\tilde{S}_j^n)_{\max}} \right) \quad (117)$$

where the maximum wave speed estimates in each spatial direction are

$$\left(\tilde{S}_i^n\right)_{\max} = \max_{i,j} \left(|u_{i,j}| + c_{i,j} \right) \quad (118)$$

and

$$\left(\tilde{S}_j^n\right)_{\max} = \max_{i,j} \left(|v_{i,j}| + c_{i,j} \right). \quad (119)$$

The Courant coefficient, C , is introduced in Equation (117) to ensure stability as described in Section E of Chapter 4.

C. Dimensional Splitting

We solve our two-dimensional IBVP approximately using a numerical technique called dimensional splitting. We choose this because its implementation is straightforward, building directly upon our one-dimensional code. In dimensional splitting, we *split* the original IBVP into two one-dimensional IBVPs

$$X \text{ IBVP} = \begin{cases} \bar{U}_t + \bar{F}(\bar{U})_x = \bar{0} \\ \bar{U}(x, y, t) \end{cases} \quad (120)$$

and

$$Y \text{ IBVP} = \begin{cases} \bar{U}_t + \bar{G}(\bar{U})_y = \bar{0} \\ \bar{U}(x, y, t) \end{cases}. \quad (121)$$

In implementing dimensional splitting, our goal is to develop a second-order accurate solution procedure that remains computationally efficient. To do so, we use the Beam and Warming variant (1976):

$$\bar{U}^{n+2} = [X^{\Delta t} Y^{\Delta t} X^{\Delta t}] (\bar{U}^n). \quad (122)$$

$X^{\Delta t}$ is the x -flow time step operator that operates on the initial data, $\bar{U}_{i,j}^n$, to obtain an interim solution, $(\bar{U}_{i,j}^{n+1})'$, by solving J , one-dimensional problems in the x -direction. Similarly, we define $Y^{\Delta t}$ as the y -flow time step operator.

To illustrate how dimensional splitting is implemented via Beam and Warming, we consider the following scenario. Given $\bar{U}_{i,j}^n$, Equation (50) is solved over the interval $[t^n, t^{n+2}]$ to obtain the second order-accurate solution, $\bar{U}_{i,j}^{n+2}$. Before we can begin our one-dimensional sweeps we must first set our boundary conditions and calculate a stable time step. Note from Equation (122) that Δt remains constant over the entire time interval. This is necessary so that first-order errors will cancel, leaving only second-order errors.

Once we have determined Δt , we begin the first of two full time steps with $X^{\Delta t}$. During this step, $X^{\Delta t}$ operates on $\bar{U}_{i,j}^n$ by solving J one-dimensional problems in the x direction. Once the interim solution, $(\bar{U}_{i,j}^{n+1})'$, is obtained, $Y^{\Delta t}$ operates on it solving I one-dimensional problems in the y direction to obtain $\bar{U}_{i,j}^{n+1}$. At this point we have completed one full time step. As a precautionary measure, we verify that the Courant-Friedrichs-Lewy stability requirement has been met. If our choice of Δt was

inappropriate, we update C , throw away our intermediate calculations and re-execute the first time sweep.

Once a stable solution is obtained at t^{n+1} , we begin the second time step over the interval $[t^{n+1}, t^{n+2}]$. Because the method employed during the first time step results in an interim solution that is biased in the y direction, we now execute the second time step in reverse order to obtain the solution, i.e. $Y^{\Delta t}$ operates on $\bar{U}_{i,j}^{n+1}$ followed by $X^{\Delta t}$ on $(\bar{U}_{i,j}^{n+2})'$ to obtain $\bar{U}_{i,j}^{n+2}$. Given the symmetry of Equation (122) and a constant Δt , this results in a cancellation of the first-order error terms giving us the second-order accuracy we desire. Before we move forward in time, we again verify the stability of the solution. Once we are satisfied that $\bar{U}_{i,j}^{n+2}$ is the proper solution, we update our simulation clock by $2 \Delta t$.

The dimensional splitting technique is repeated as many times as required to obtain the solution at the desired time as shown in Figure 28. It is important to note that the dimensional splitting technique used here is inherently parallel. Calculations performed during each time step operator may be executed independently. With future parallel implementation in mind, we wrote the code using Fortran 90 and Fortran 95 language extensions. Subroutines and functions were written to be pure and independent. To make the code parallel, all that remains is to add the required High Performance Fortran language extensions.

While this method is simple, the two-dimensional nature of our time step operators complicates the solution procedure. In the next section, we address these special considerations.

```

Set  $t = 0$ 
Set initial conditions
Do until  $t \geq T$ 
  Set  $\bar{U}_{old} = \bar{U}$ 
  Set  $\bar{W}_{old} = \bar{W}$ 
  Set boundary conditions in each phantom cell (in order to)
  Calculate maximum wave speeds in  $x$  and  $y$  directions
  Do time step calculations until a stable second-order solution is obtained
    Calculate  $\Delta t$ 
    Do  $x$  direction one-dimensional problems independently for  $j = 1$  to  $J$ 
      For all cell interfaces,  $i = -1$  to  $I + 1$ , solve the Riemann problems
      For all cell interfaces,  $i = 0$  to  $I$ , calculate the limited WAF
      For all cells,  $i = 1$  to  $I$ , update  $\bar{U}$  and  $\bar{W}$ 
    End Do
    Set boundary conditions in  $y$  phantom cells
    Do  $y$  direction one-dimensional problems independently for  $i = 1$  to  $I$ 
      For all cell interfaces,  $j = -1$  to  $J + 1$ , solve the Riemann problems
      For all cell interfaces,  $j = 0$  to  $J$ , calculate the limited WAF
      For all cells,  $j = 1$  to  $J$ , update  $\bar{U}$  and  $\bar{W}$ 
    End Do
  End Do

```

Figure 28: Pseudocode for 2D Code

If $v_{\max}^1 < 1$ the first time step calculation is stable
 Set boundary conditions in y phantom cells
 Do y direction one-dimensional problems independently for $i = 1$ to I
 For all cell interfaces, $j = -1$ to $J + 1$, solve the Riemann problems
 For all cell interfaces, $j = 0$ to J , calculate the limited WAF
 For all cells, $j = 1$ to J , update \bar{U} and \bar{W}
 End Do
 Set boundary conditions in x phantom cells
 Do x direction one-dimensional problems independently for $j = 1$ to J
 For all cell interfaces, $i = -1$ to $I + 1$, solve the Riemann problems
 For all cell interfaces, $i = 0$ to I , calculate the limited WAF
 For all cells, $i = 1$ to I , update \bar{U} and \bar{W}
 End Do
 If $v_{\max}^2 < 1$ the second time step calculation is stable
 Exit second-order time step loop
 Else
 Update Courant coefficient
 Set $\bar{U} = \bar{U}_{old}$
 Set $\bar{W} = \bar{W}_{old}$
 Start second-order time step over
 End If
 Else
 Update Courant coefficient
 Set $\bar{U} = \bar{U}_{old}$
 Set $\bar{W} = \bar{W}_{old}$
 Start second-order time step over
 End If
 End Do
 Update simulation time: $t = t + 2\Delta t$
 End Do

Figure 28 continued

D. Special Considerations

Recall from Equation (8), that in two dimensions \vec{U} is a four-component vector: ρ (mass), ρu (x momentum), ρv (y momentum) and E (total energy). Therefore, special care must be taken in solving our one-dimensional sweeps. To remain directionally independent, we define \vec{W} and \vec{U} in terms of normal and tangential components, i.e.

$$\vec{W} = (\rho, V_N, V_T, p) \quad (123)$$

and

$$\vec{U} = (\rho, P_N, P_T, E) \quad (124)$$

where V_N , V_T , P_N , and P_T represent the normal and tangential components of velocity and momentum respectively. So, for the general one-dimensional problem, we solve a system of four PDEs.

Recall that, given m PDEs, we have m waves in the Riemann problem structure. Therefore, in two dimensions, the general Riemann problem wave structure in the one-dimensional WAF consists of four waves: wave 1 (left outer wave), wave 2a (contact surface), wave 2b (tangential velocity shear wave) and wave 3 (right outer wave) as shown in Figure 29. We denote the inner waves as wave 2a and wave 2b to reinforce the fact that they travel at the same speed and are therefore indistinguishable in the $x - t$ plane.

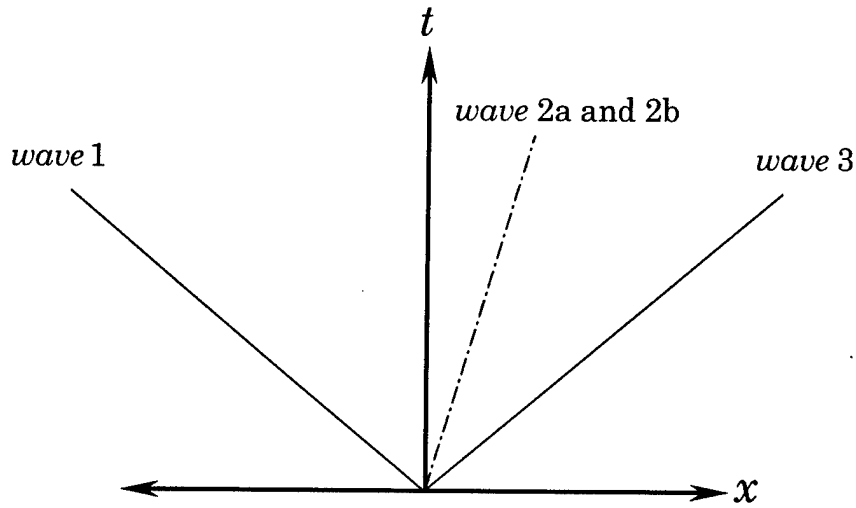


Figure 29: WAF Wave Structure in Two-Dimensions

When we solve the Riemann problem via Godunov's method, the solution procedure for the tangential components is trivial. When we apply limited WAF however, the solution procedure becomes more complicated. Recall that in limited WAF, each wave, w_k , is amplified by its corresponding wave amplifier, α_k , which is dependent upon r_k . Typically, $r_{2a} \neq r_{2b}$. This results in a dispersed Riemann problem wave structure that complicates the solution procedure. In Figure 30, we see that the application of limited WAF results in four waves and five regions.

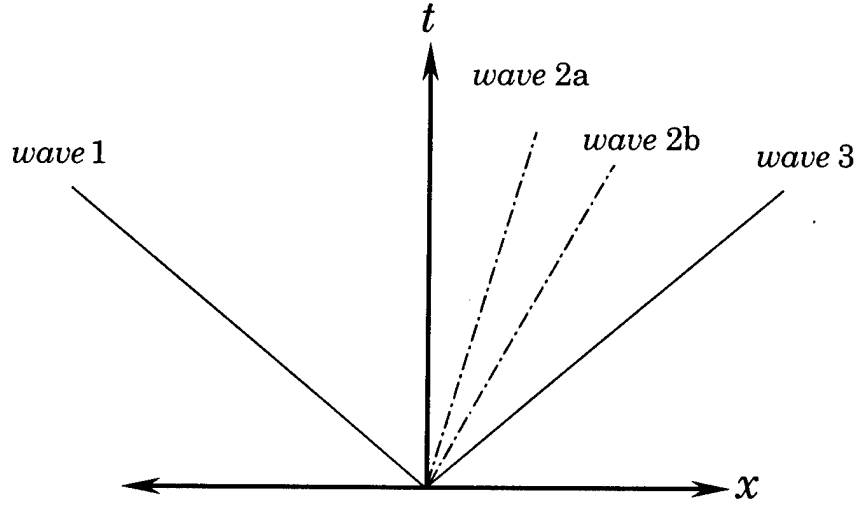


Figure 30: Limited WAF Wave Structure in Two-Dimensions

We calculate the limited weighted average flux, \bar{F}^{LWAF} , as a two step process where the normal and tangential components are calculated separately. To illustrate how this is done, we consider a one-dimensional calculation in the x direction. As described in Chapter 4, we calculate \bar{F}^{LWAF} from the weighted average state, \bar{W}^{LWAF} . To calculate $\rho_{i+1/2}^{LWAF}$, $V_{N,i+1/2}^{LWAF}$ and $p_{i+1/2}^{LWAF}$, we simply ignore the tangential velocity shear wave, w_{2b} , and solve the problem as previously described. The flux of ρ and ρV_N is then calculated directly.

Notice that ρV_T and E , and their associated fluxes, are dependent upon the tangential velocity component, V_T . We calculate $V_{T,i+1/2}^{LWAF}$ by solving a hypothetical Riemann problem separately that consists of a single

wave, wave 2b, and two constant state regions, $V_{T,1}$ and $V_{T,4}$. The tangential velocity is calculated by (Toro, 1997):

$$V_{T,i+1/2}^{LWAF} = \bar{w}_5 V_{T,1} + \bar{w}_6 V_{T,4}. \quad (125)$$

The limited weights, \bar{w}_5 and \bar{w}_6 , are defined as (Toro, 1997):

$$\bar{w}_5 = \frac{1}{2}(1 + \phi_{2b}) \quad (126)$$

and

$$\bar{w}_6 = \frac{1}{2}(1 - \phi_{2b}) \quad (127)$$

where $\phi_{2b} = \alpha_{2b} v_{2b}$. Unlike our typical TVD weight limiter functions

$(\phi_1, \phi_{2a}, \phi_3)$, ϕ_{2b} is dependent upon the tangential velocity gradient and not the density gradient, i.e.

$$r_{i+1/2,2b} = \frac{\Delta V_{T,i+1/2-\sigma,2b}}{\Delta V_{T,i+1/2,2b}}. \quad (128)$$

We now have a solution procedure for every component of \bar{W}^{LWAF} and

\bar{F}^{LWAF} . No other special considerations are required.

E. Sod Test in Two-Dimensions

Before we look at more complex problems, we solved the Sod problem in two-dimensions to test our code. The two-dimensional Sod test was performed over a square computational spatial domain, $[0,1] \times [0,1]$. To maintain the one-dimensional nature of the problem we applied reflective

boundaries at $y = 0$ and $y = 1$. Each parameter was set to the same values as before. We applied the van Leer A TVD weight limiter function to achieve monotonic-like results near discontinuities. The following initial conditions were used:

ICs :

$$\bar{W}(x, y, 0) = \begin{cases} \bar{W}_1 = (1.0 \text{ kg/m}^3, 0.0 \text{ m/s}, 0.0 \text{ m/s}, 1.0 \text{ Pa})^T & x < 0.5 \text{ m} \\ \bar{W}_4 = (0.125 \text{ kg/m}^3, 0.0 \text{ m/s}, 0.0 \text{ m/s}, 0.10 \text{ Pa})^T & x > 0.5 \text{ m} \end{cases} \quad (129)$$

Figure 31 shows our two-dimensional results. These are in agreement with

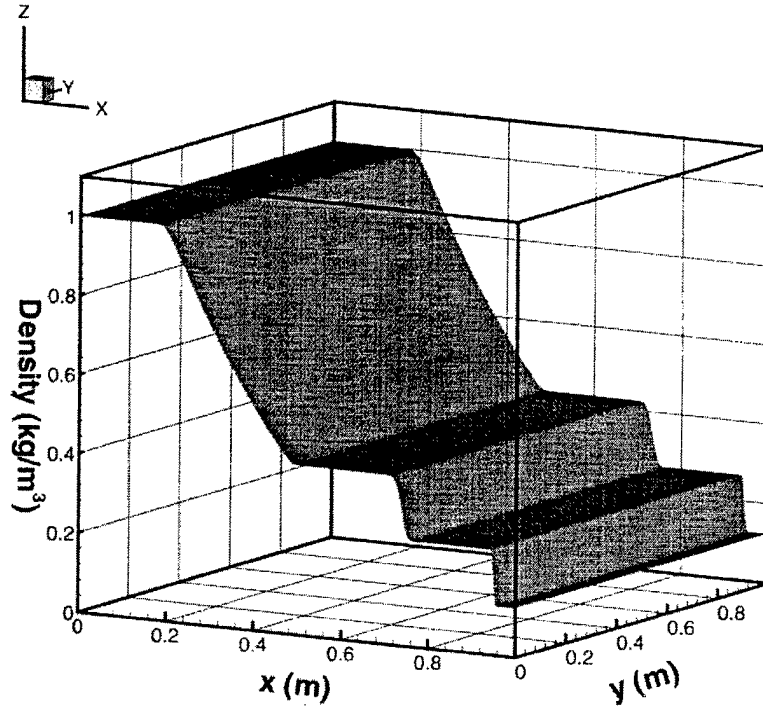


Figure 31: 2D Sod Density Profile at $t=0.25s$

those obtained in one dimension. In Sod's two-dimensional problem, flows occur in only one direction. To ensure the code models two component flows correctly, we solved a series of cylindrical explosion problems as presented by

Toro (1997). In each case, we placed a high-pressure cylindrical region at the center of a square computational domain. The solution of each was verified in the following manner:

- primitive variable profiles were qualitatively compared against Toro's results at planes $\theta = 0, \pi/4, \pi/2, 3\pi/4$ and π as measured from the axis to the leading edge of the shock front.
- the cylindrical symmetry of the solution was verified.

Chapter 6: High Pressure Cylinder Problem

In Chapter 5 we extended our one-dimensional shock code into two dimensions using dimensional splitting but only gave a cursory look at the numerical capabilities of the code. Here we further investigate the numerical properties and capabilities of the code to solve complex problems of interest to DoD; namely, cylindrical shock wave propagation and reflection over ideal surfaces.

A. Problem

To illustrate the numerical capabilities of our two-dimensional shock code, we simulate the complicated shock phenomena that occur from the detonation of a cylindrically symmetric explosive 4.0m above an ideal surface. In our model, we completely ignore the detonation process itself and concern ourselves with its mechanical effects only; the outward propagation of the resulting cylindrical shock wave. To simulate the near instantaneous deposition of energy that occurs and the resulting high pressures immediately after detonation, we introduce a hypothetical cylinder normal to the $x - z$ plane centered at $(x, z) = (0m, 4m)$ with $r = 0.25m$. The ambient air within the cylinder is heated until we reach three times the ambient pressure, i.e. $p_{Cyl} = 3 p_0$. This gives us an infinitely long high-pressure cylinder parallel to the ideal surface that:

- is the physical analogue of an annular shock tube, i.e. removal of the cylindrical diaphragm begins the simulation.
- produces a cylindrical shock wave shortly after removal of the diaphragm similar to that produced by the explosive.
- is simple to model numerically.

B. Shock Wave Propagation and Reflection over an Ideal Surface

Before solving the problem numerically, we discuss the behavior of the shock wave as it propagates outward and reflects over an ideal surface. The qualitative discussion below provides the necessary background to understand and interpret the numerical results presented in Section C. For a more detailed discussion of shock reflection phenomena we suggest the following books: Gabi Ben-Dor's Shock Wave Reflection Phenomena, Samuel Glasstone and Philip Dolan's The Effects of Nuclear Weapons and Gilbert Kinney's Explosive Shocks in Air.

Shock Wave Propagation

We begin our discussion just after the detonation of the explosive at $t = 0$ s as modeled by our high-pressure cylinder approximation. The simulation begins with the instantaneous removal of the cylindrical diaphragm. With the removal of our fictitious barrier, the heated air is no longer constrained and begins to accelerate rapidly outward down the pressure gradient. When the particle velocity of the heated gas exceeds the ambient speed of sound, a cylindrical shock front is formed that propagates

radially outward. As the shock front continues its outward expansion, cylindrical divergence of the flow results in a decreasing peak overpressure at the front, i.e. $\Delta p^{\max} \propto 1/r$.

Directly behind the shock front, there is an outwardly expanding cylindrical contact discontinuity. As in the one-dimensional case, this is due to the difference in internal energy between the shock-heated air ahead of the contact discontinuity with the cooler expanded air behind it. In our investigation of shock propagation and reflection, the contact discontinuity is of little interest. We simply mention it here for completeness.

At the same time that the shock front and contact discontinuity are formed, a rarefaction wave begins to travel up the pressure gradient towards the axis of the high-pressure cylinder. At the instant the wave reaches the axis, it instantaneously reflects the cylindrical wave. The resulting rarefaction wave accelerates as it travels outward through the previously shock-heated air. Eventually the rarefaction head catches up to and interacts with the shock front resulting in the characteristic coupled wave structure shown in Figure 32.

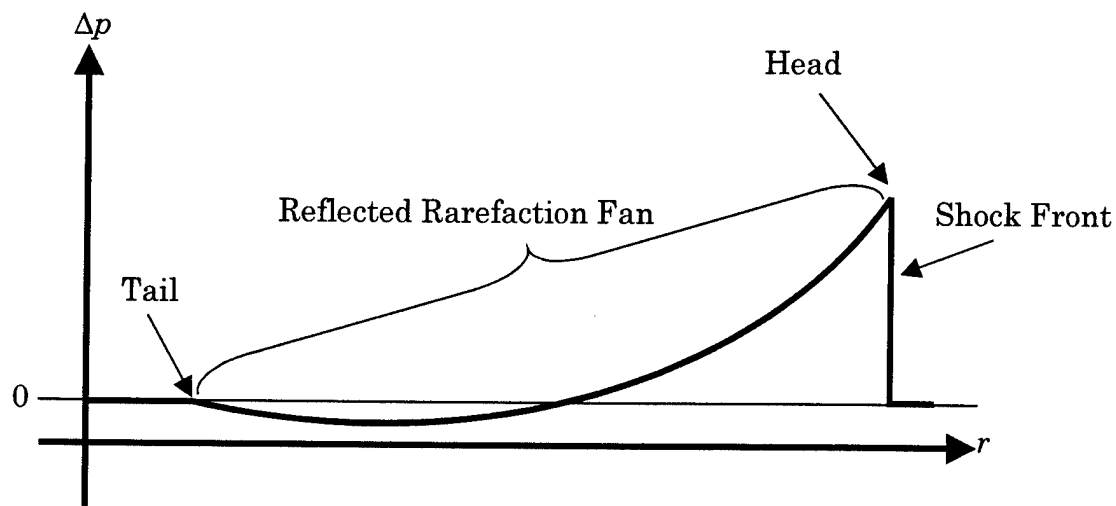


Figure 32: Typical Overpressure Profile

Notice that the decrease in pressure behind the shock front is due entirely to the interaction of the reflected rarefaction wave. In addition, the minimum overpressure within the rarefaction becomes negative at some point in time.

The cylindrical shock wave above continues to propagate as described until ambient conditions are reached. Near the surface however, the flow is far more interesting and complicated. When our initial shock wave reaches the surface, various reflection phenomena occur. We will observe the following shock reflection phenomena: normal reflection, regular reflection, transition from regular to Mach reflection, and Mach reflection.

Shock Wave Reflection

Up to now we have concerned ourselves with the formation and propagation of the shock wave over the temporal interval $[t^0, t_a)$ as shown in Figure 33. At $t = t_a$, the incident shock wave, I , is traveling perpendicular to

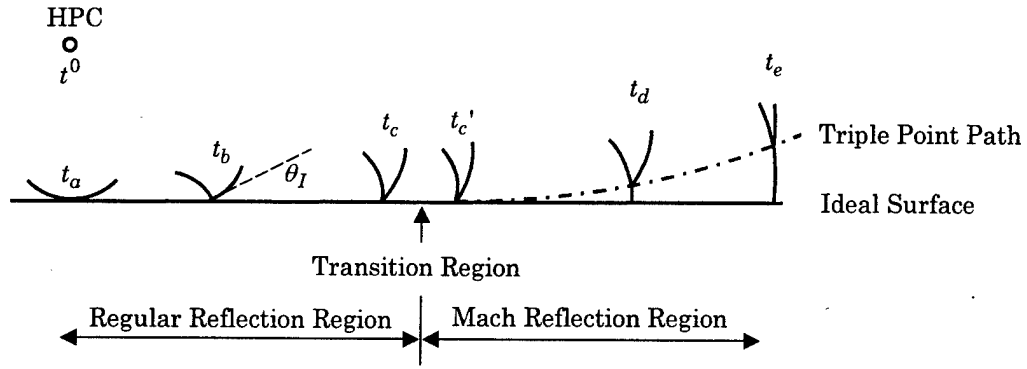


Figure 33: Typical Shock Reflection Phenomena

the surface and normal reflection occurs instantaneously at the point $x = 0m$.

When the flow is normal to the surface meaning no tangential components

are present, the resulting peak overpressure, Δp_R^{\max} , is at its maximum

possible value, Δp_R^\perp . The reflected peak overpressure at normal reflection,

Δp_R^\perp , is a least twice the incident overpressure, Δp_I^{\max} (Glasstone and Dolan,

1977):

$$\Delta p_R^\perp = 2 \Delta p_I^{\max} \frac{7p_0 + 4 \Delta p_I^{\max}}{7p_0 + \Delta p_I^{\max}} \quad (130)$$

where p_0 is the ambient pressure before detonation.

As the incident shock wave continues to propagate, e.g. $t_a < t < t_c$, its direction of travel becomes oblique with respect to the surface. Immediately after $t = t_a$, a reflected shock wave (R) is formed marking the end of normal reflection. The resulting two-shock configuration, diagramed in Figure 34, is called regular reflection.

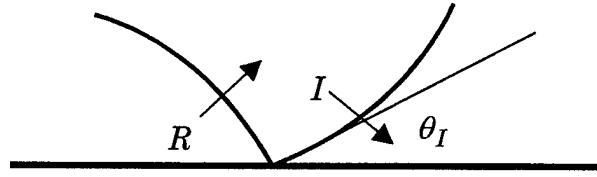


Figure 34: Regular Reflection

With reference to Figure 34, P is the reflection point where wave I and R intersect at the surface. The peak overpressure, Δp_R^{\max} , occurs at this point due to the stagnation of the normal components of the flow behind I as reflection occurs. As I continues to propagate over the surface, the angle of incidence, θ_I , increases resulting in a corresponding decrease in the normal flow component. In addition, Δp_I^{\max} decreases as described earlier due to cylindrical divergence. Therefore Δp_I^{\max} decreases and is always less than Δp_R^{\max} .

Regular reflection continues until a critical angle, θ_I^{crit} , is achieved. In Figure 33, this occurs at $t = t_c$. At this point, we begin the transition from a two-shock regular reflection to a three-shock configuration called Mach reflection. This occurs as R begins to coalesce into I forming a single shock front called the Mach stem (M). This occurs as the shock moves through the transition region, during the interval $t_c < t < t_c'$.

Mach reflection begins when the Mach stem is formed as depicted in Figure 35. The peak overpressure at the Mach stem is greater than that observed for either I or R , i.e. $\Delta p_M^{\max} > \Delta p_R^{\max} > \Delta p_I^{\max}$. As we move up M , the overpressure decreases. The intersection of M , R and I is called the triple point (T). Notice in Figure 33 that for $t > t_c'$, the height of T increases over time as M continues to grow.

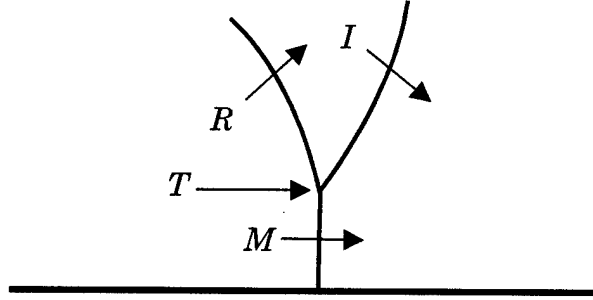


Figure 35: Mach Reflection

Now that we have an understanding of shock propagation and reflection, we present and discuss the numerical results obtained using our two-dimensional shock code in the next section. If our two-dimensional code models the physics of inviscid flow in two-dimensions correctly and our implementation is sound, we should observe the shock phenomena described above.

C. Numerical Solution

To model the complicated shock phenomena that results from the detonation of our cylindrical explosive we solved the two-dimensional IBVP given the ICs in Equation (131) and BCs in Table 5. Recall the high-

$$\bar{W}(r,0) = \begin{cases} \bar{W}_1 = (1.2045 \text{ kg/m}^3, 0.0 \text{ m/s}, 0.0 \text{ m/s}, 304.05 \times 10^3 \text{ Pa})^T & r < 0.25m \\ \bar{W}_4 = (1.2045 \text{ kg/m}^3, 0.0 \text{ m/s}, 0.0 \text{ m/s}, 101.35 \times 10^3 \text{ Pa})^T & r > 0.25m \end{cases} \quad (131)$$

x BCs	z BCs
Reflective along $x = 0m$.	Reflective along $z = 0m$.
Transmissive along $x = 20m$.	Transmissive along $z = 12m$.

Table 5: High-pressure Cylinder Boundary Conditions

pressure cylinder models the energy deposition immediately after detonation and is centered at $(x,z) = (0m, 4m)$. To reduce the execution time of the code, we took advantage of the symmetry of the problem and inserted a reflective boundary along the $x = 0m$ plane. Our remaining reflective boundary simulated the ideal surface, i.e. the ground. To restrict our computational domain to a finite size, we employed transmissive boundaries at $x = 20m$ and $z = 12m$. Each was positioned at a sufficient distance to allow observation of interesting flow features while delaying the arrival of the inevitable false reflected waves until after the passage of the shock through the region of interest. These boundary conditions therefore define the following spatial domain: $[0m, 20m] \times [0m, 12m]$.

Initially, we subdivided our spatial domain into 600×360 square computational cells. The square cell geometry was used, instead of the more typical and computationally efficient rectangular cell, to limit the numerical artifacts that occur in approximating our cylinder on a Cartesian mesh. In addition, we used a weighted average of the initial conditions in diaphragm

cells. As we shall see in our results, this significantly decreased the stair-step-like numerical artifacts that occur at early times resulting in a nice cylindrical wave structure.

Lastly, we set the execution parameters as shown in Table 6.

C_0	C_{\max}	Q_S	Tolerances	TVD Weight Limiter
0.2	0.9	2	1.0×10^{-9}	van Leer A

Table 6: High-pressure Cylinder Execution Parameters

Notice the Courant coefficient is set as described in Chapter 4 to maintain stability at early times and ϕ_{VLA} was chosen to ensure sharp resolution of the shocks of interest.

We ran the simulation over the time interval $[0ms, 60ms]$. To ensure we captured regular reflection, transition from regular reflection to Mach reflection, and Mach reflection as described in Section B, we recorded the location and value of the maximum overpressure every other time step as shown in Figure 36- 38. In addition, the overpressure solution was saved every $0.4ms$. This data was then used to visualize the structure of the shock waves at times of interest using Tecplot (ver 7.5), a high-end scientific visualization software package.

During our discussion of the numerical results we will follow the maximum overpressure curves shown in Figure 36, 37 and 38. These curves serve to guide our discussion and show the progress of the simulation as the cylindrical shock wave forms, propagates outwards and reflects over the

surface. Periodically, we will look at the overpressure solution obtained at various times as well.

In Figure 36, it is immediately obvious from our time plot that two time intervals exist: a free-air time interval where the shock has not yet reached the surface and an interval over which reflection occurs. During the free-air interval, $[0ms, 9.84ms)$, the maximum overpressure occurs along the shock front of the cylindrical shock wave as it propagates outwards. We found that Δp^{\max} occurs at the shock front and decreases like $1/r$ due to cylindrical divergence.

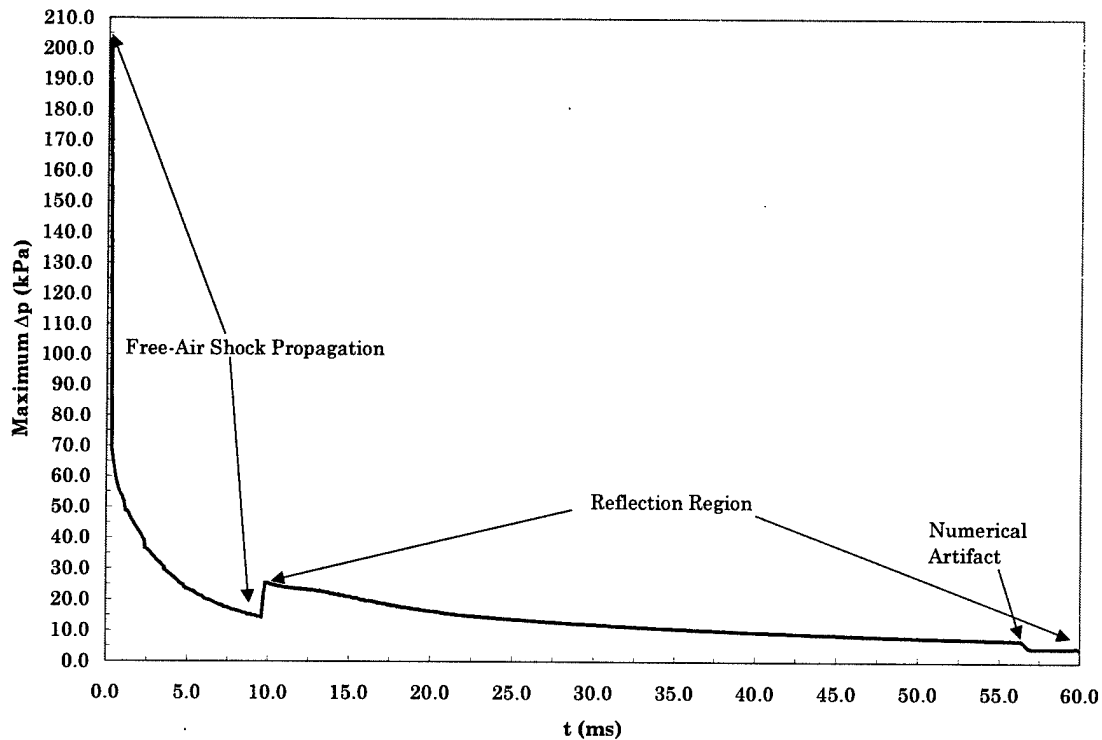


Figure 36: Maximum Overpressure vs Time $[0ms, 60ms]$

In Figure 39, we see that the high-pressure cylinder approximation has produced a nearly cylindrical shock wave as desired. The stair-case-like wave patterns present along the shock front are numerical artifacts of our weighted cylinder approximation and cannot be helped. A finer mesh will limit this effect considerably but at the significant cost of additional computational effort.

At $t \cong 9.84ms$, we observed a sharp rise in the maximum overpressure from $14.29kPa$ to $25.35kPa$ as shown in Figure 36. This indicates that the cylindrical shock wave has reached the surface and reflected producing the peak overpressure observed at the reflection point on the surface. Note that this rise in overpressure is inconsistent with Equation (130) because we did not capture the exact moment at which normal reflection occurred.

Figure 37 illustrates the maximum overpressure versus time over the time interval $[9.84ms, 60ms]$. During this time interval reflection occurs and the maximum overpressure remains located at the surface. At early times where $\theta_I < \theta_I^{crit}$, regular reflection occurs as shown in Figure 40. Here we see an excellent example of regular reflection. Both the reflected and incident waves are well resolved. In addition, we note that the peak reflected overpressure occurs due to the stagnation of the incident flow at the reflection point.

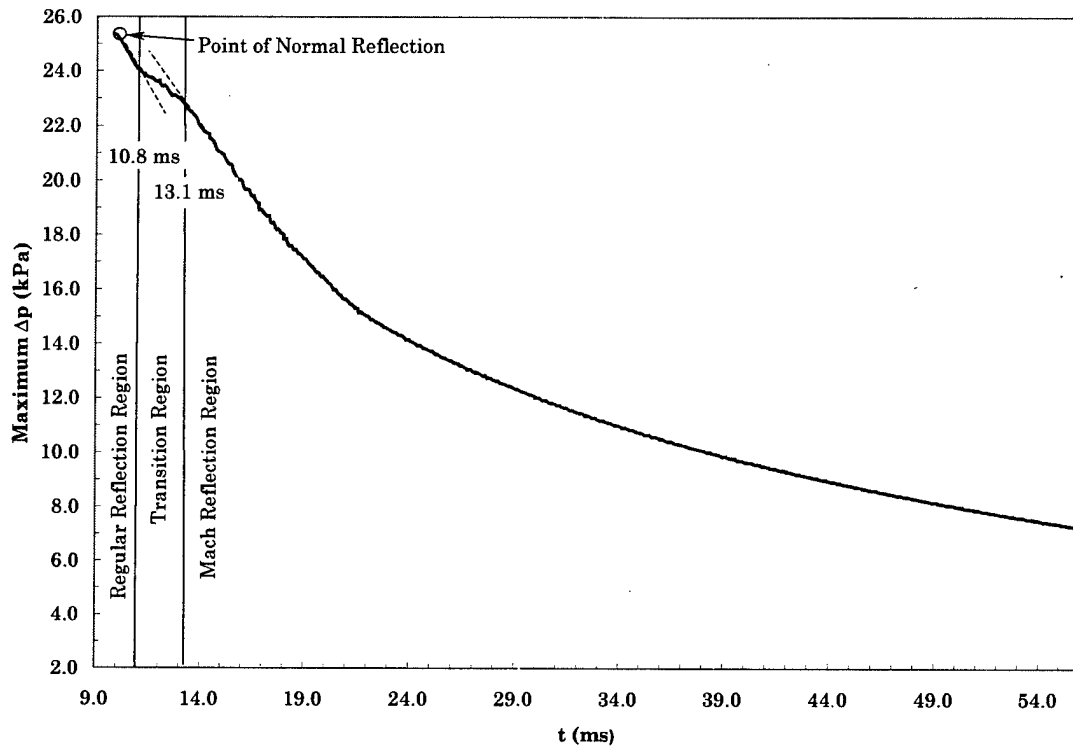


Figure 37: Maximum Overpressure vs Time [9.84ms, 60ms]

Notice that if we extend the left and right portions of the curve as depicted by the dashed lines in Figure 37 we can clearly see a bump that occurs roughly between 10.8 and 13.1ms. Over this time interval, the behavior of Δp_R^{\max} changes; it decreases at a rate slower than that observed at earlier times until $t \approx 13.1\text{ms}$. This change is due to the fact that the reflected wave is catching up to and stagnating off the incident wave, i.e. transition is occurring from regular to Mach reflection. At $t \approx 13.1\text{ms}$, transition ends with the formation of the Mach stem and Δp_M^{\max} decreases at a faster rate. Therefore, the reflection time interval may be subdivided into

three distinct intervals: $[9.84ms, 10.8ms)$ - regular reflection, $[10.8ms, 13.1ms]$ - transition and $(13.1ms, 60ms]$ - Mach reflection.

In Figure 38, we plotted the maximum overpressure during reflection versus the ground distance at which it occurred. Both are functions of t and give us the time of arrival of the shock on the ground. We subdivided the reflection interval into regions just as we did in Figure 37. Here we see that the transition region begins near $x = 1.68m$ and ends around $x = 3.28m$.

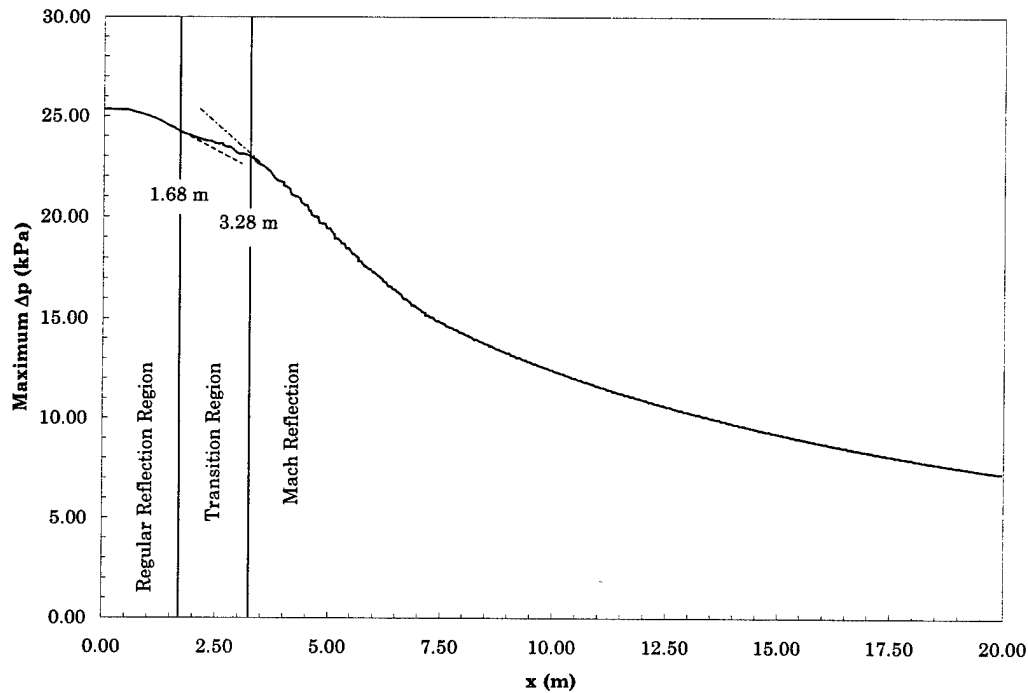


Figure 38: Maximum Overpressure vs Ground Distance

To illustrate and investigate the transition from regular to Mach reflection, we refined the mesh by a factor of two and obtained the overpressure solutions every $0.4ms$ over the interval $[10.4ms, 13.6ms]$ as

shown in Figure 42 - 49. As we proceed in time through the transition region, we see that as θ_I increases the distance at the surface between the reflected and incident shock waves decreases steadily. In addition, the width of the peak overpressure region decreases but grows in height suggesting that flows behind the reflected wave are beginning to stagnate as it catches up to the incident wave. This indicates that the reflected and incident shock waves are beginning to merge near the surface. By $t = 12.8ms$, we see the beginnings of a Mach stem forming near the ground and by $t = 13.6ms$ a Mach stem is present.

In Figure 51 – 52 we see the well-resolved three-shock configuration of Mach reflection. As time goes on and the waves continually merge we observe the growth of the Mach stem, the motion of the triple point away from the surface and the decrease in the peak overpressure. In addition, we notice that as we move up the Mach stem the stagnation pressure decreases as we expect.

Just before $t = 58ms$ we see in Figure 37, 37 and 38 that the peak overpressure at the Mach stem decreases suddenly. This is a numerical artifact of the right transmissive boundary. We observe the unexpected dip in peak overpressure because a false reflected shock wave has traveled back to the left and perturbed the solution. The dip indicates the leading edge of that false reflected shock wave.

From our qualitative analysis we see that the two-dimensional shock code is capable of modeling the complicated shock phenomena of interest to DoD.

23.57
22.94
22.31
21.67
21.04
20.41
19.77
19.14
18.51
17.87
17.24
16.61
15.97
15.34
14.71
14.07
13.44
12.81
12.17
11.54
10.91
10.28
9.64
9.01
8.38
7.74
7.11
6.48
5.84
5.21
4.58
3.94
3.31
2.68
2.04
1.41
0.78
0.14
-0.49
-1.12
-1.75
-2.39
-3.02
-3.65
-4.29
-4.92
-5.55
-6.19
-6.82
-7.45

Peak Overpressure = 24.20 kPa

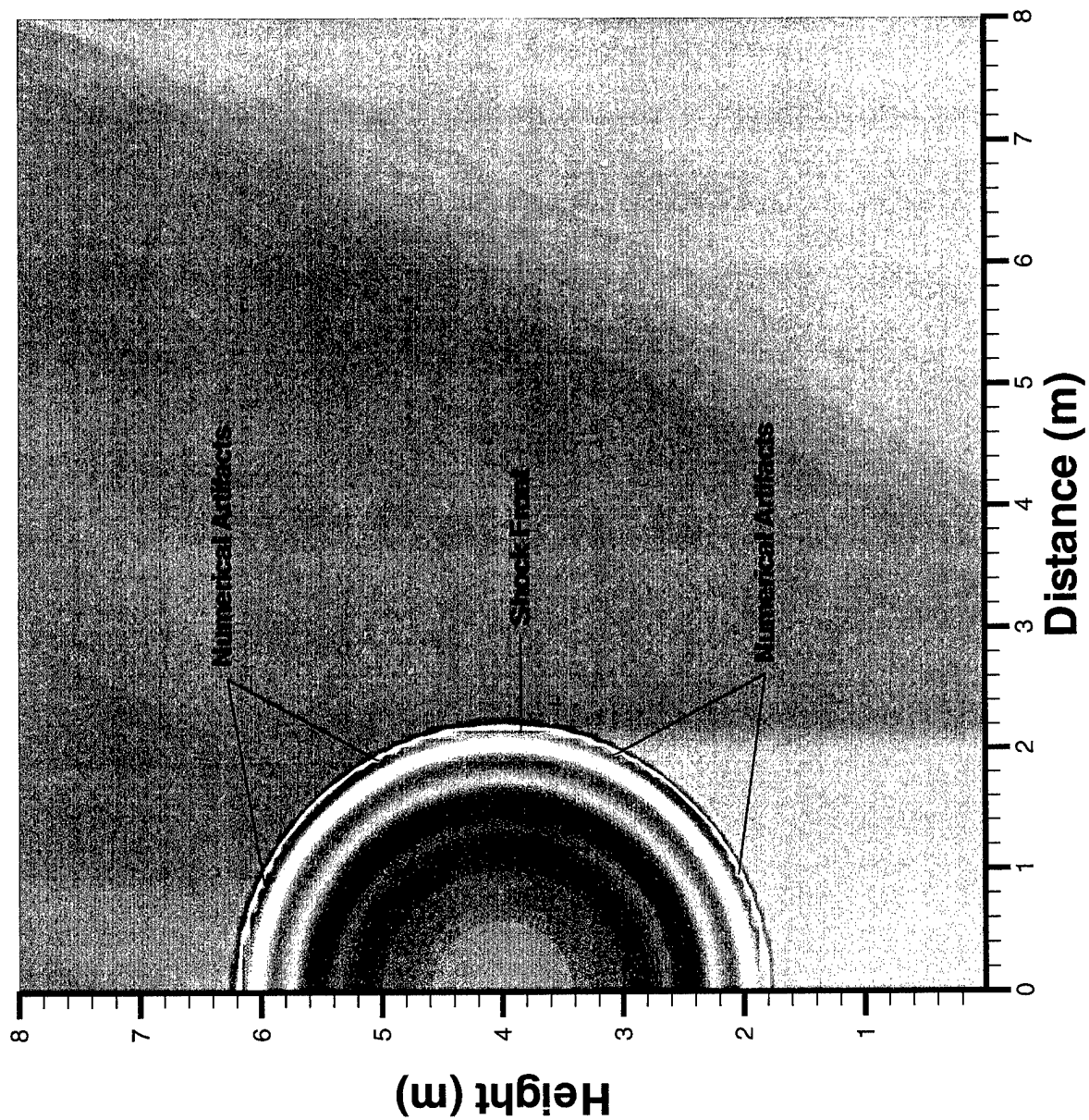


Figure 39: High-pressure Cylinder Overpressure at $t=4.8ms$

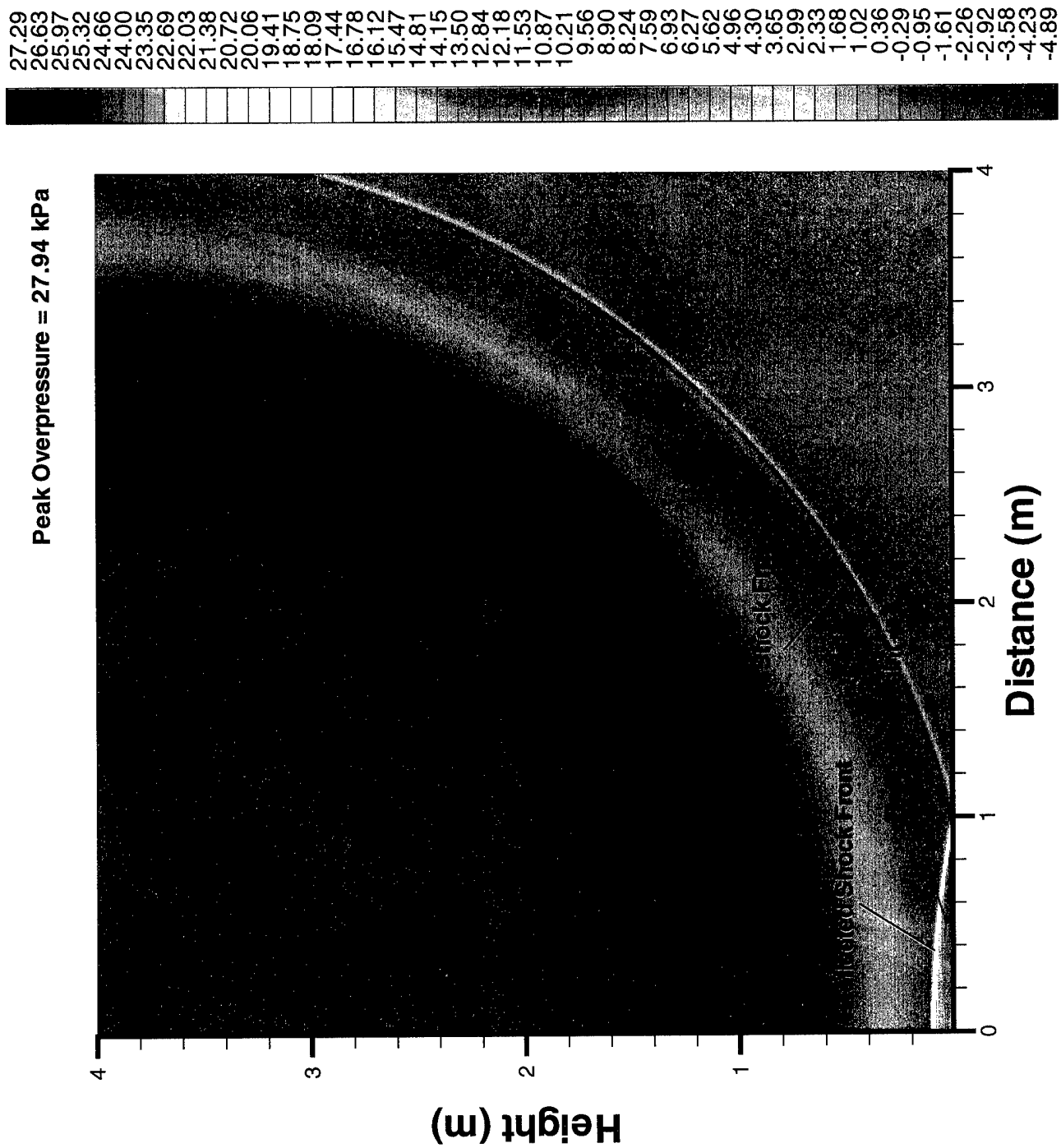


Figure 40: High-pressure Cylinder Overpressure at $t=10.0ms$

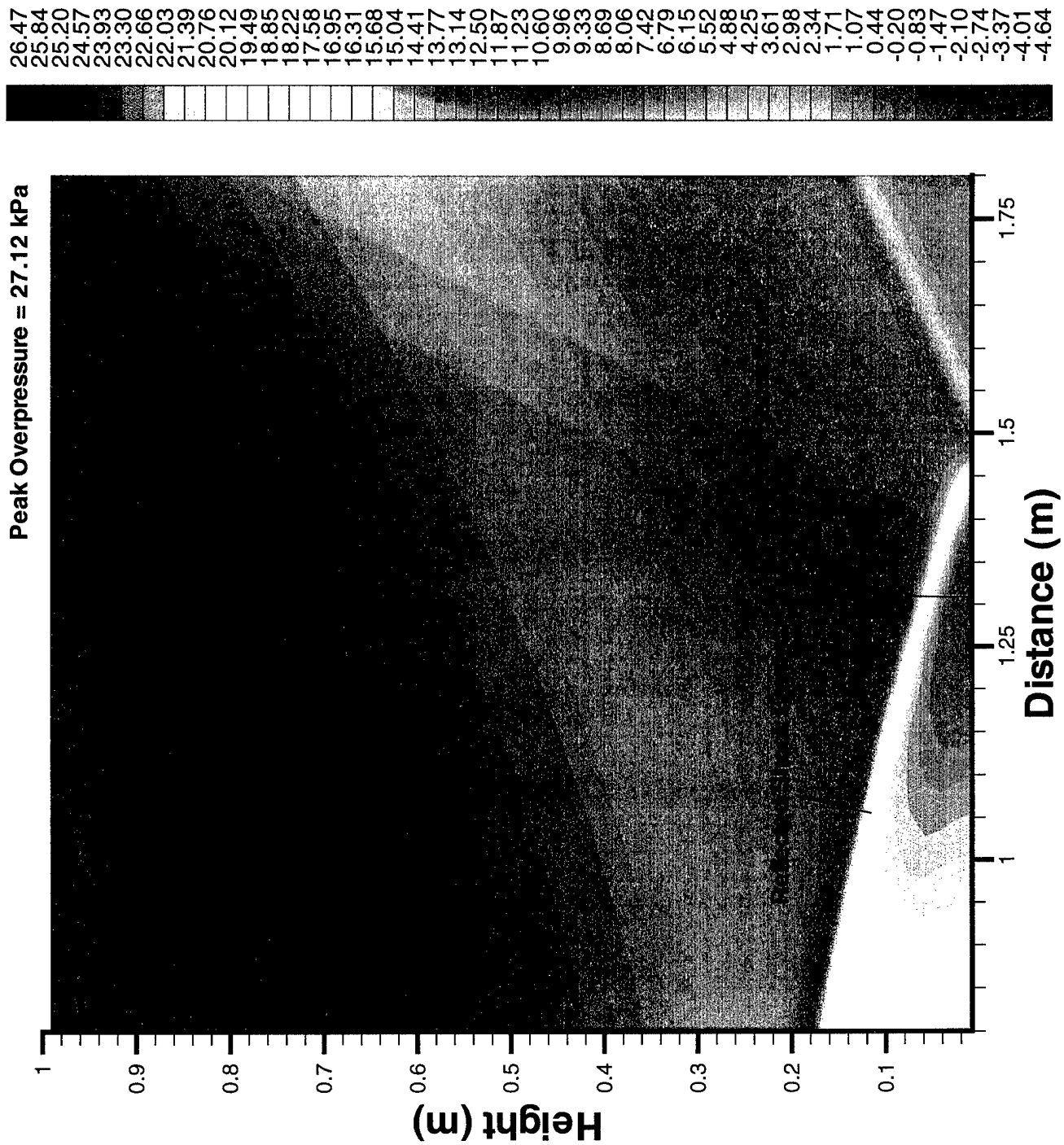


Figure 41: High-pressure Cylinder Overpressure at $t=10.4ms$

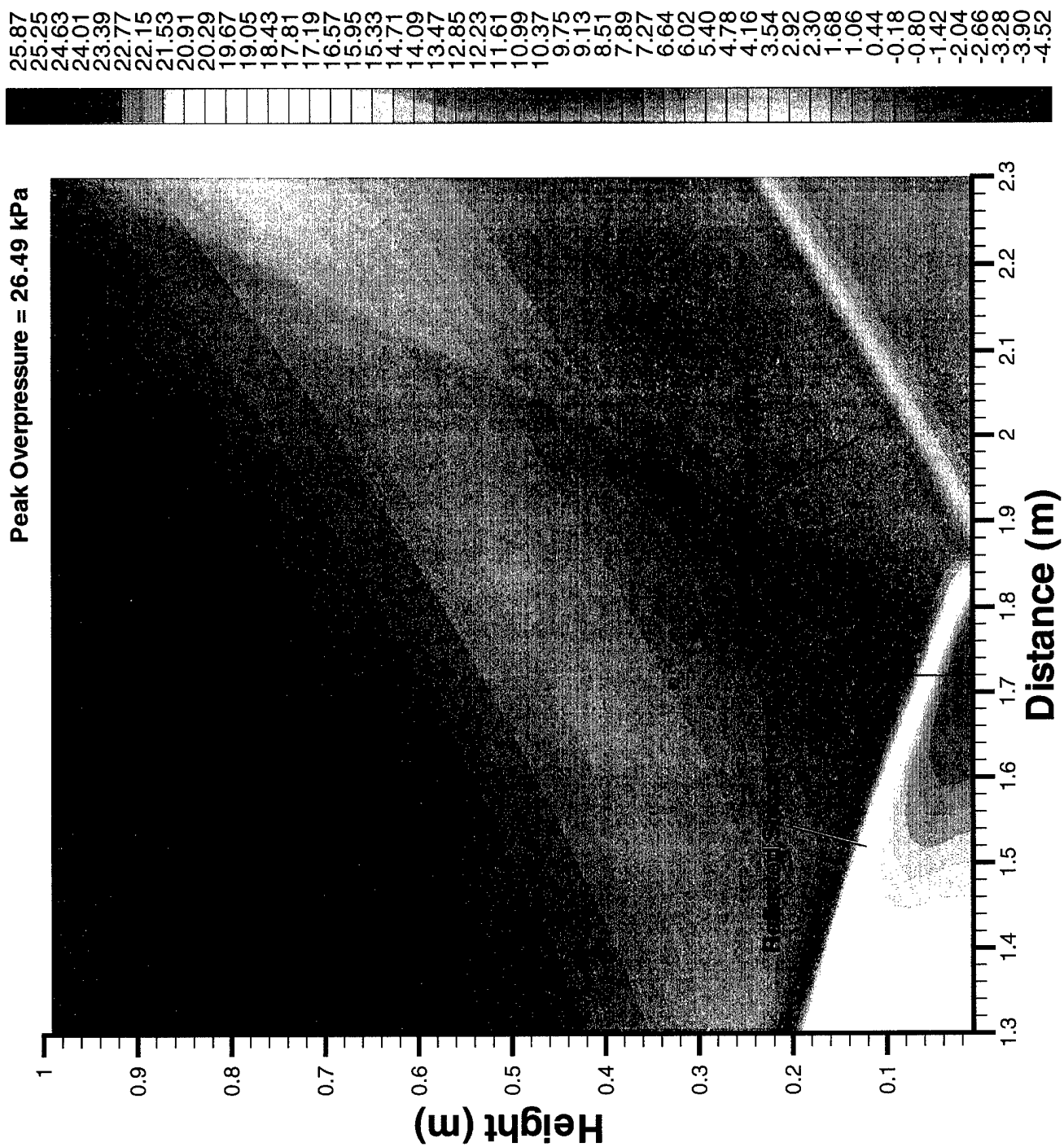


Figure 42: High-pressure Cylinder Overpressure at $t=10.8ms$

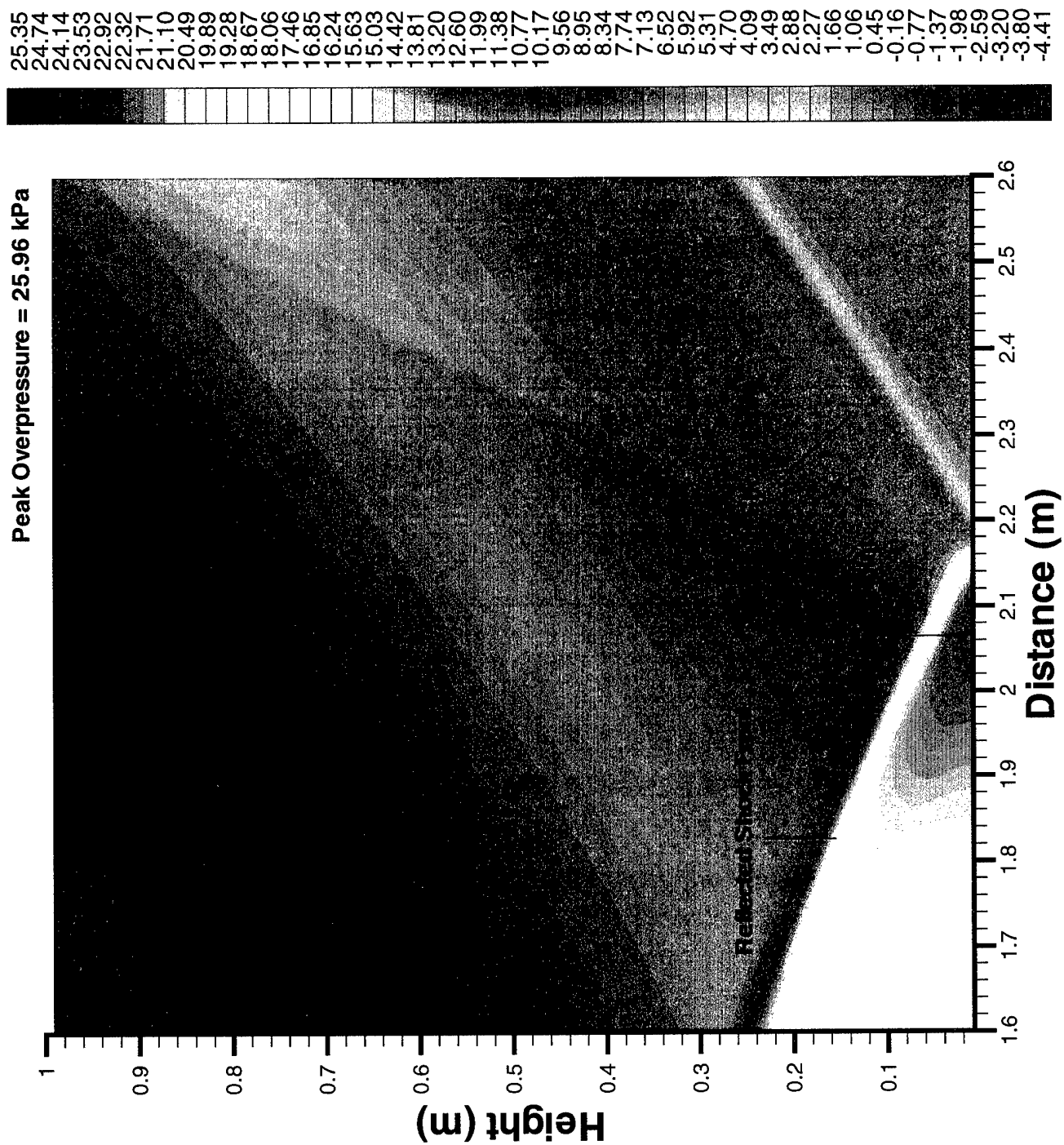


Figure 43: High-pressure Cylinder Overpressure at $t=11.2ms$

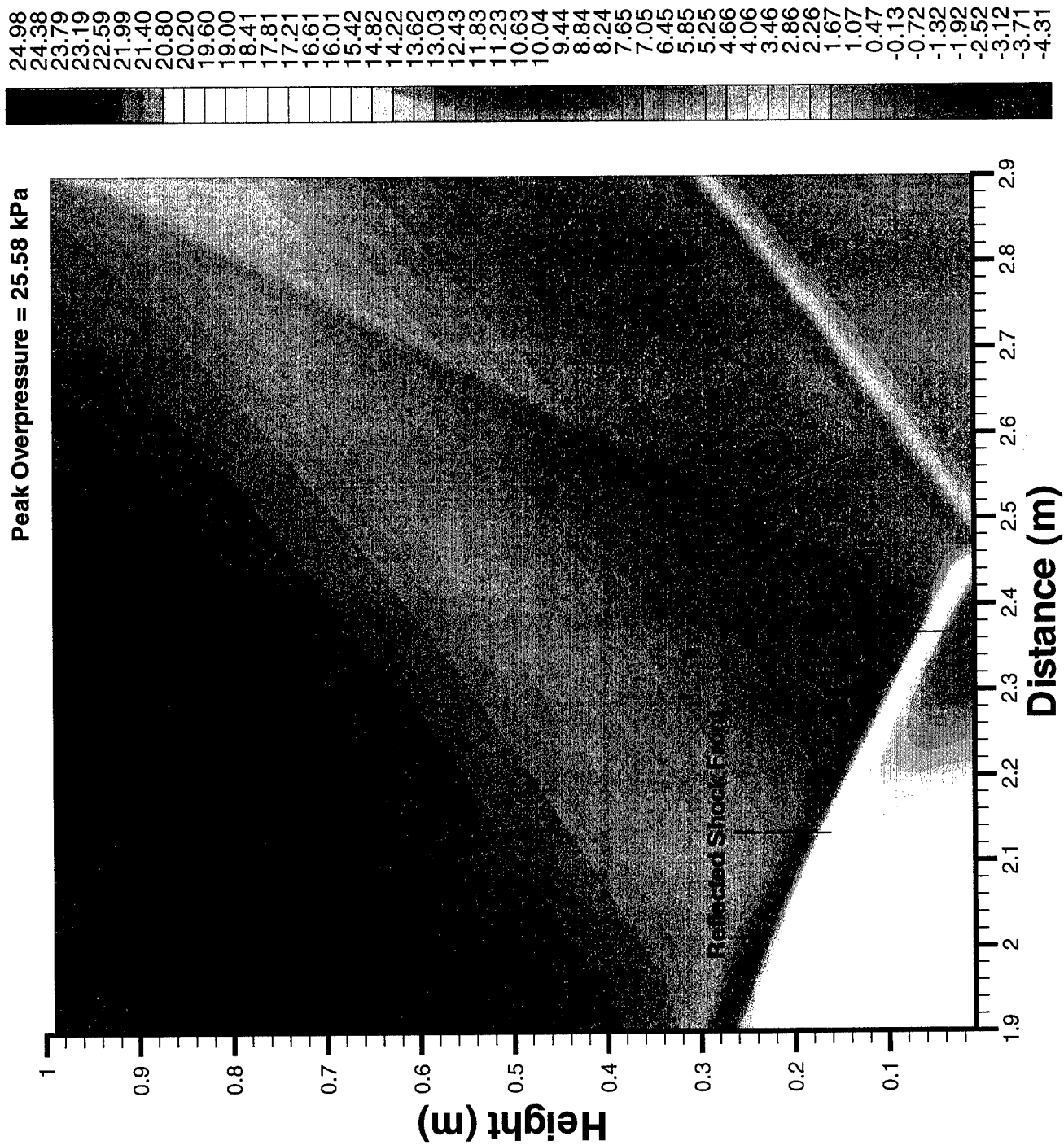


Figure 44: High-pressure Cylinder Overpressure at $t=11.6ms$

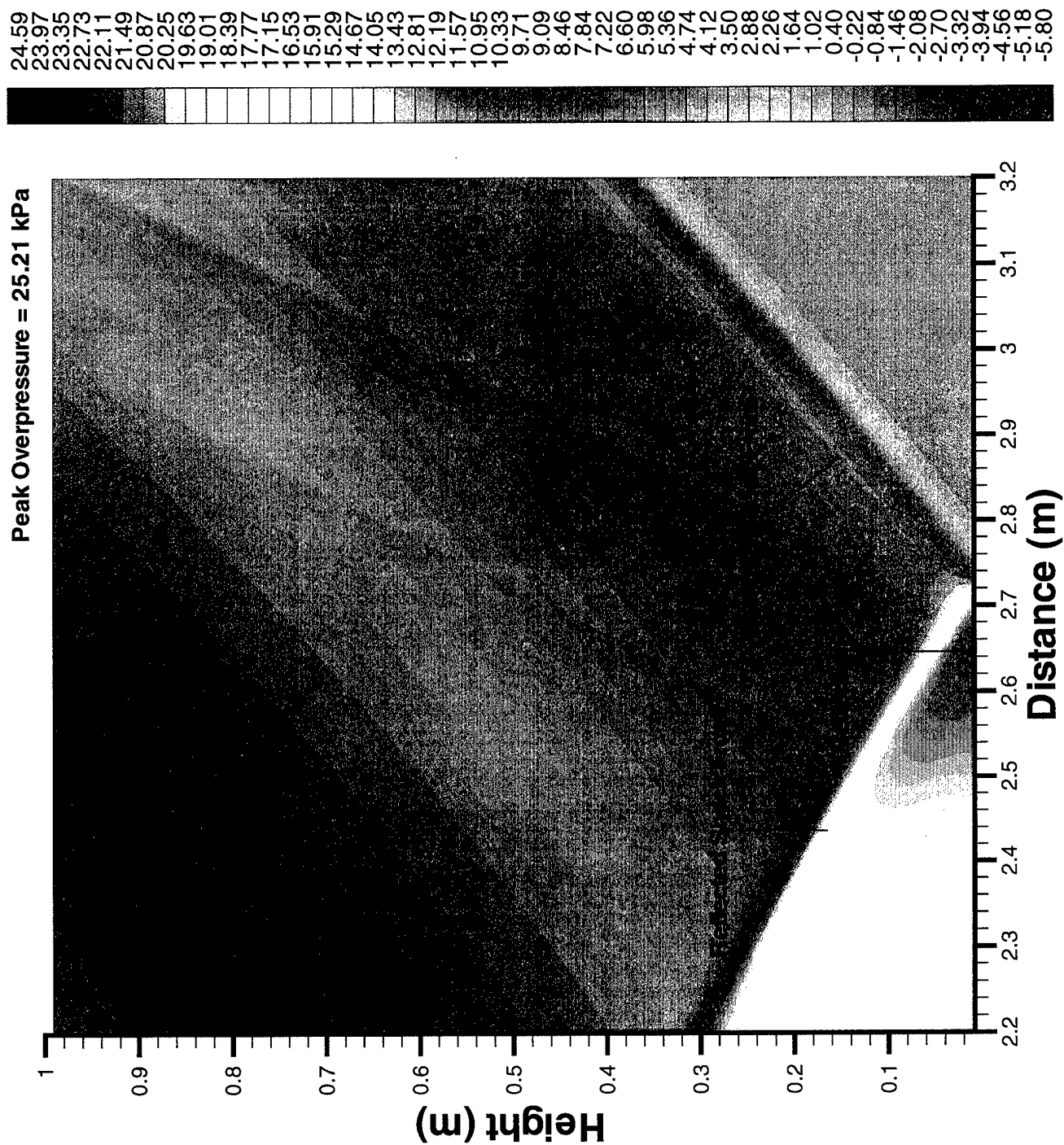


Figure 45: High-pressure Cylinder Overpressure at $t=12.0ms$

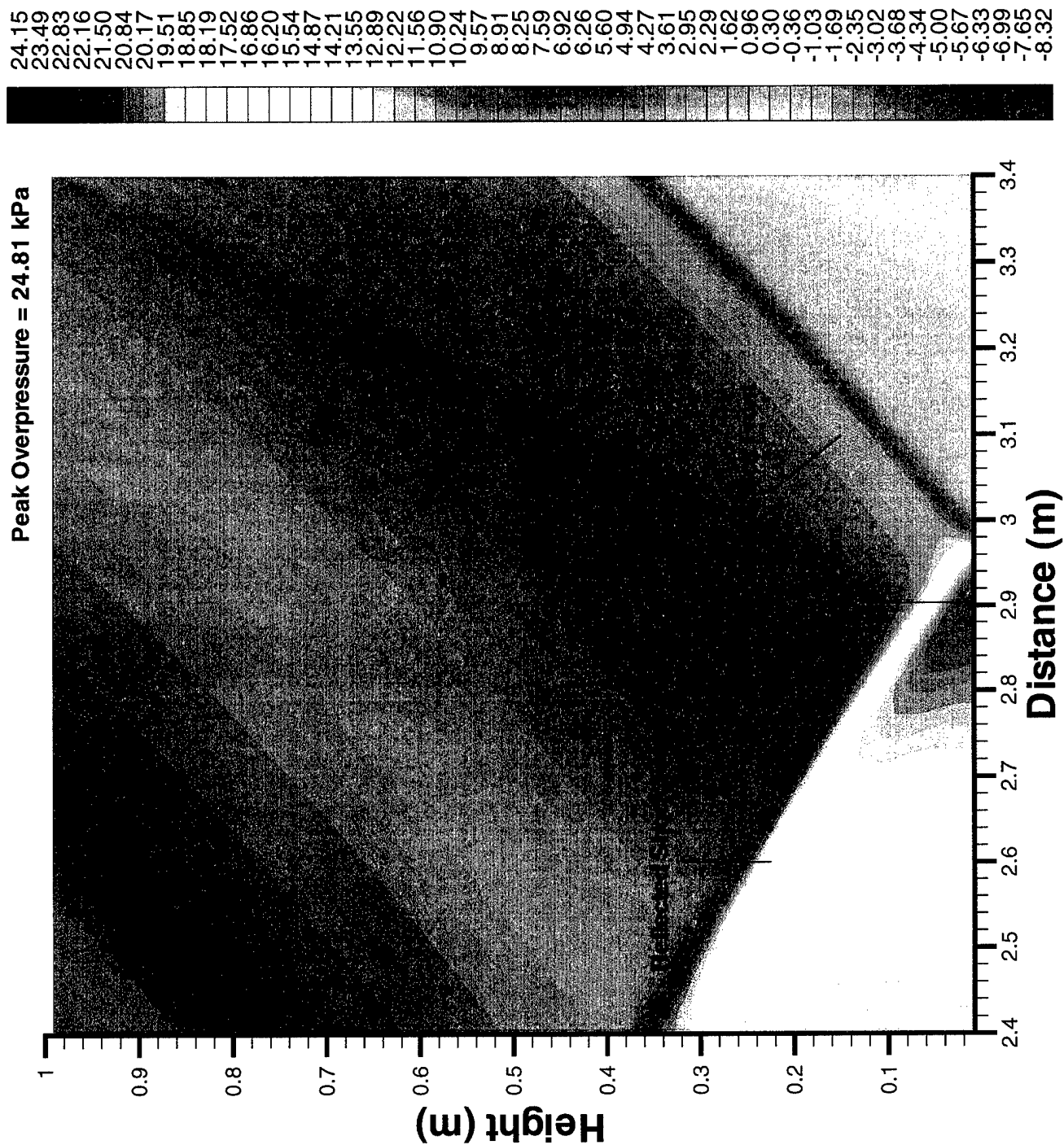


Figure 46: High-pressure Cylinder Overpressure at $t=12.4ms$

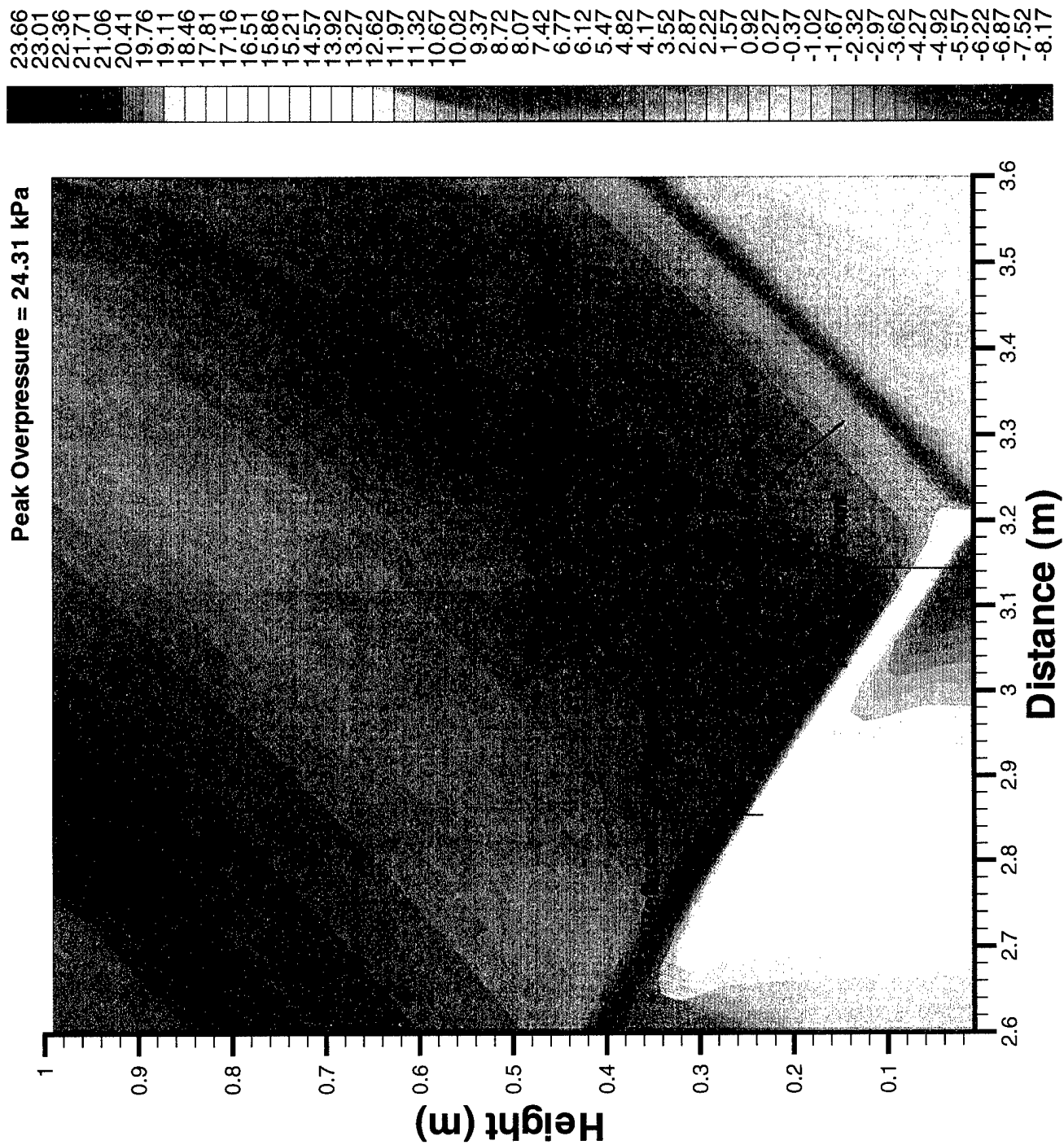


Figure 47: High-pressure Cylinder Overpressure at $t=12.8ms$

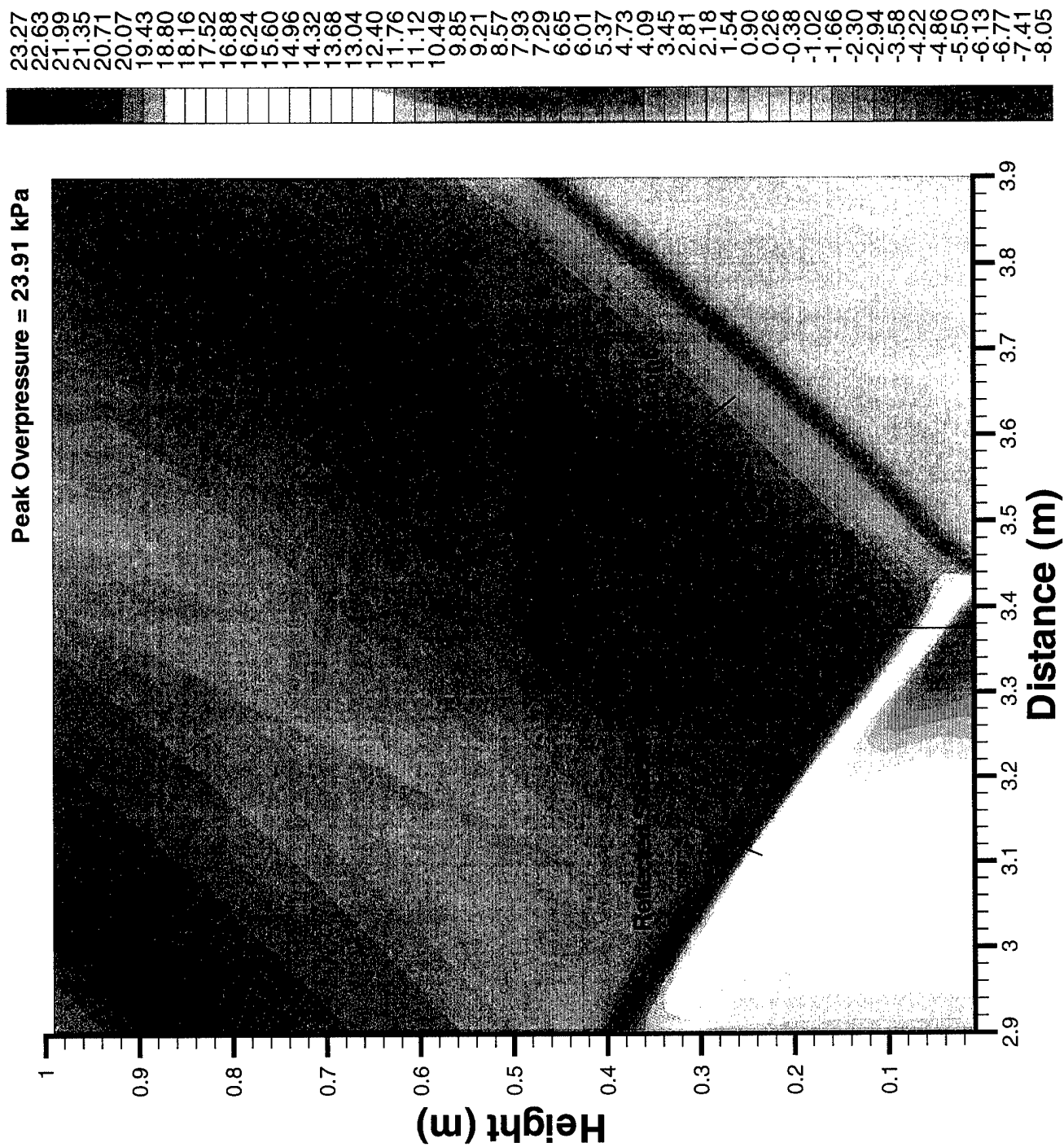


Figure 48: High-pressure Cylinder Overpressure at $t=13.2ms$

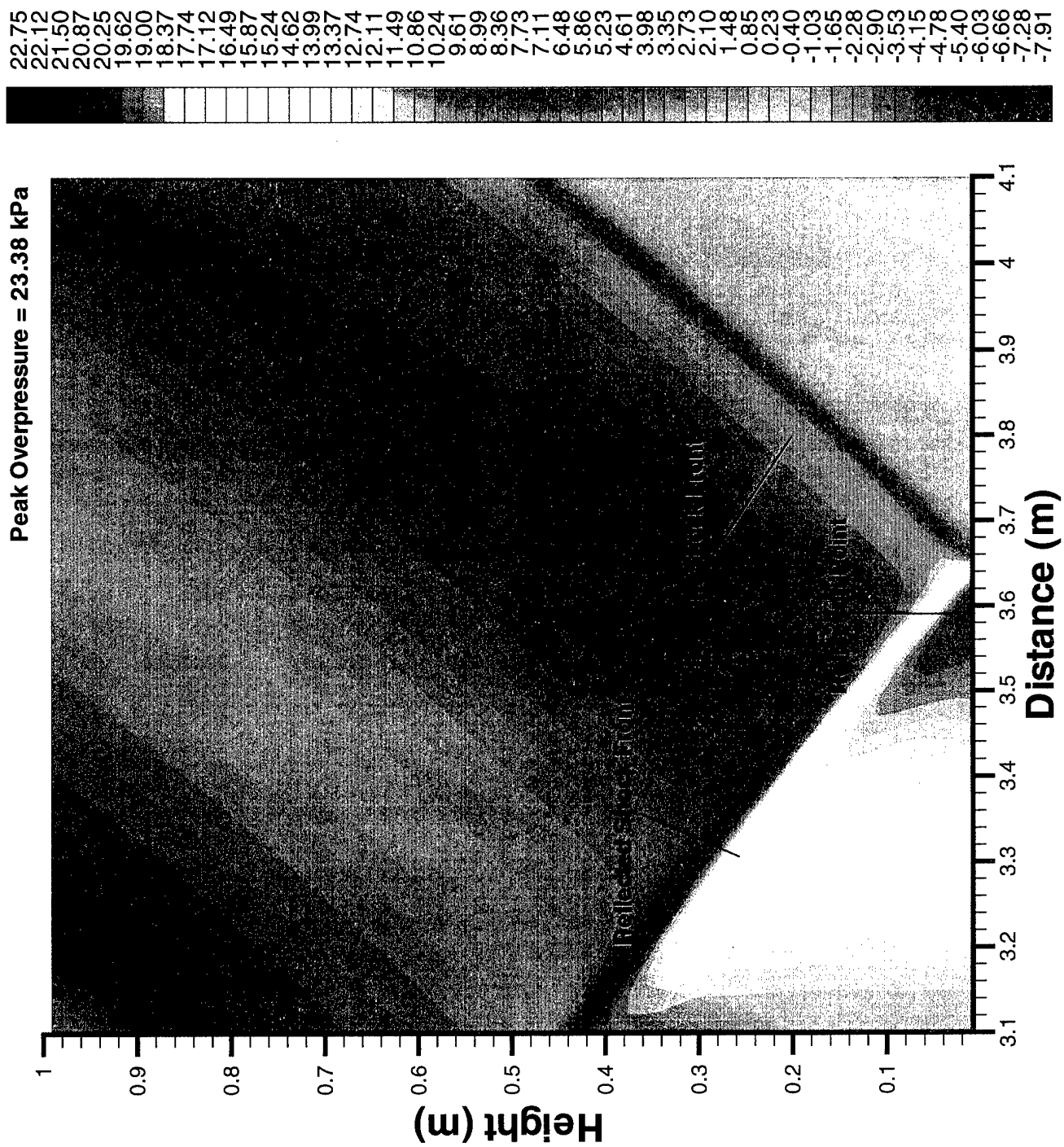


Figure 49: High-pressure Cylinder Overpressure at $t=13.6ms$

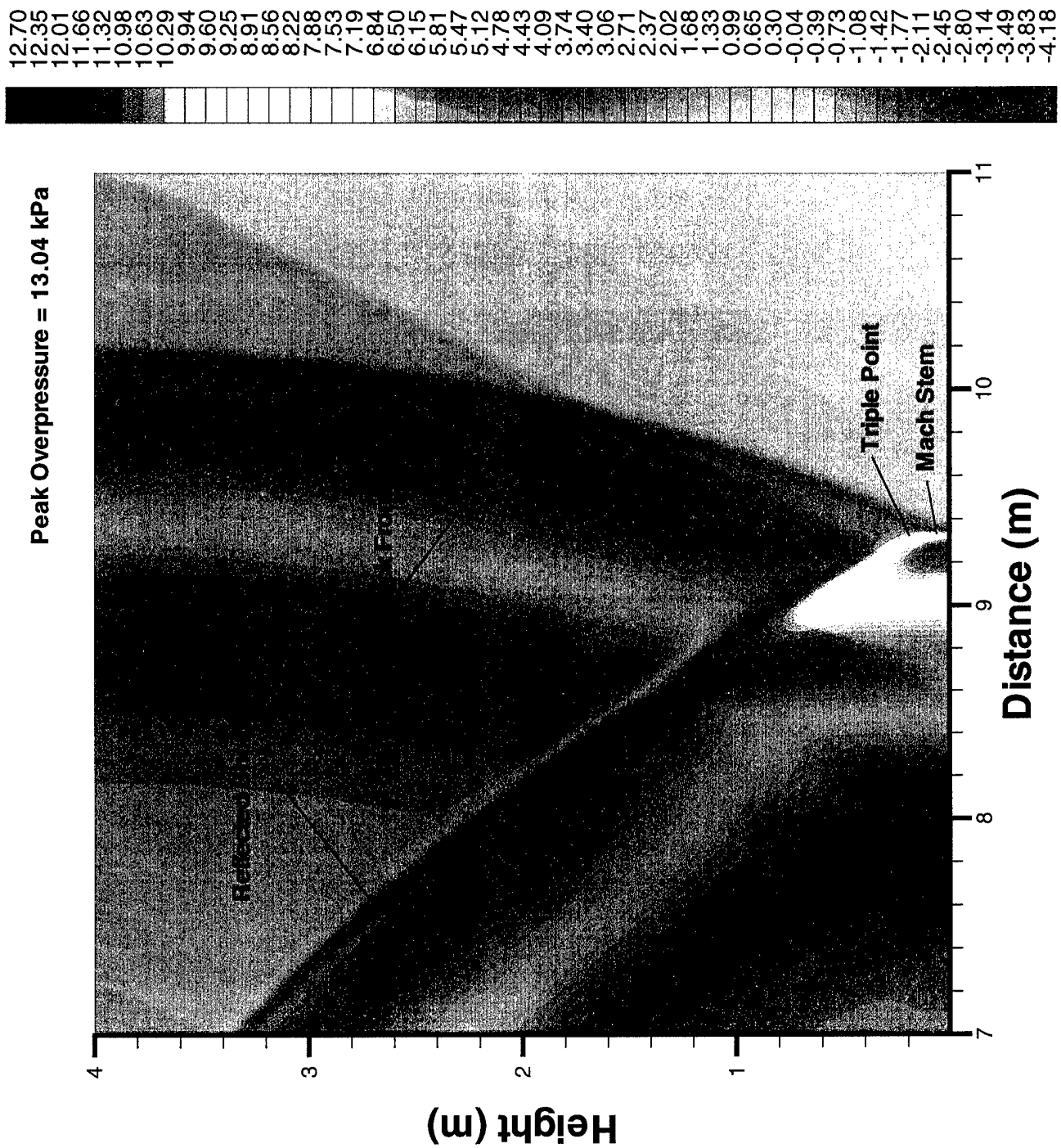


Figure 50: High-pressure Cylinder Overpressure at $t=27.0ms$

11.44
11.13
10.82
10.52
10.21
9.90
9.59
9.28
8.98
8.67
8.36
8.05
7.74
7.44
7.13
6.82
6.51
6.21
5.90
5.59
5.28
4.97
4.67
4.36
4.05
3.74
3.43
3.13
2.82
2.51
2.20
1.90
1.59
1.28
0.97
0.66
0.36
0.05
-0.26
-0.57
-0.88
-1.18
-1.49
-1.80
-2.11
-2.41
-2.72
-3.03
-3.34
-3.65

Peak Overpressure = 11.75 kPa

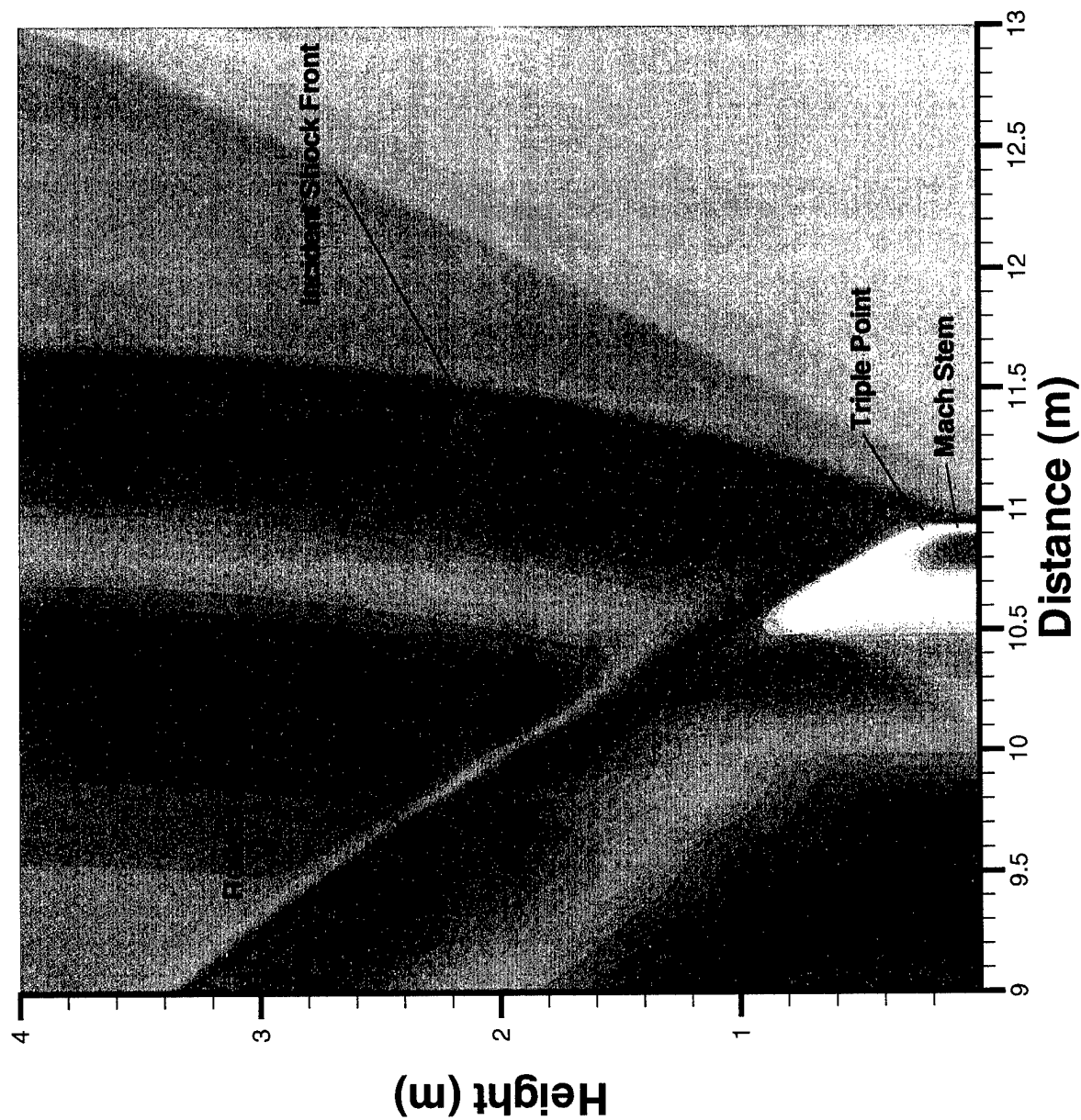


Figure 51: High-pressure Cylinder Overpressure at $t=31.2\text{ms}$

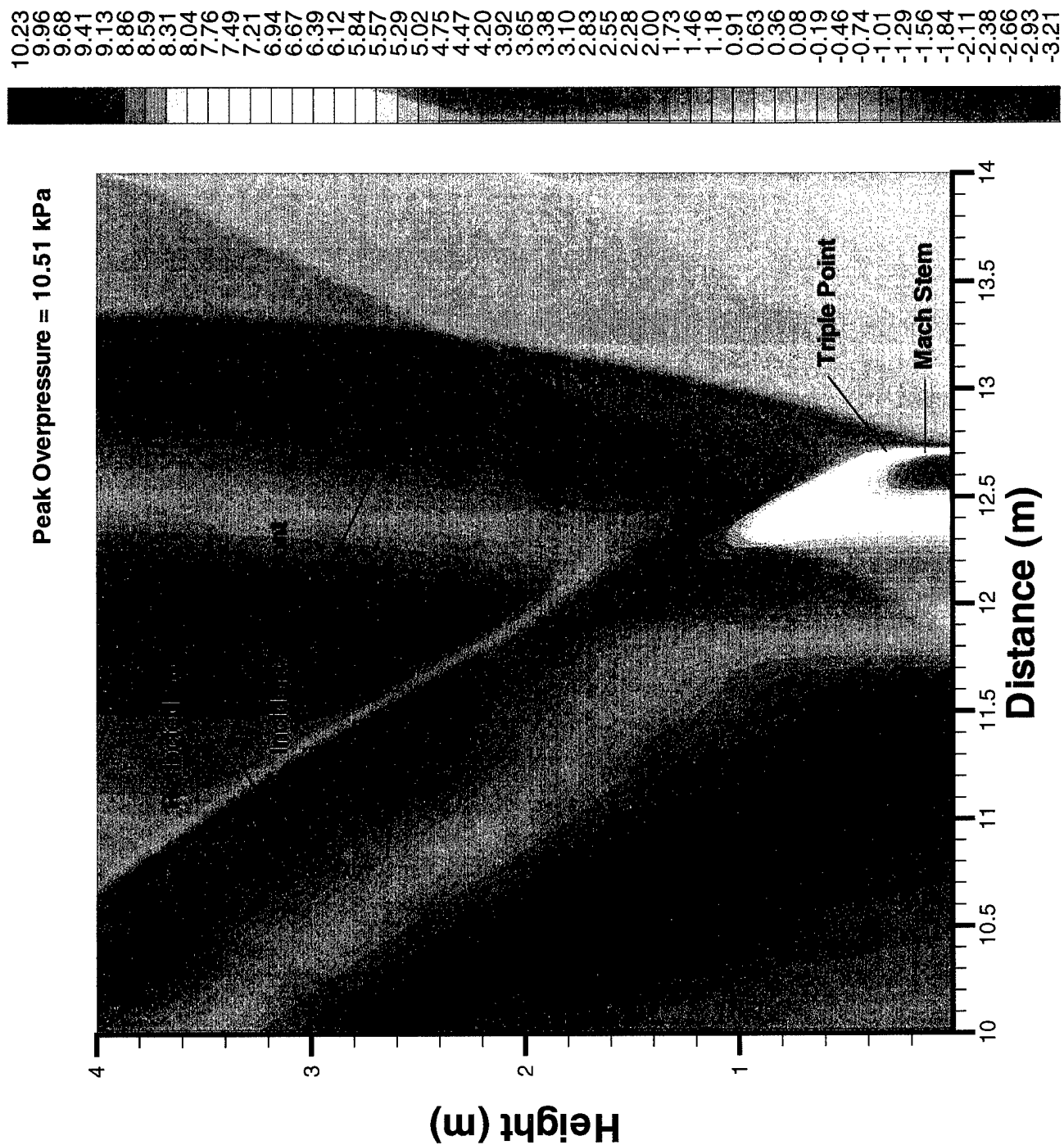


Figure 52: High-pressure Cylinder Overpressure at $t=36.0ms$

Chapter 7: Conclusion

The primary objective of this effort has been to develop a two-dimensional hydrodynamic shock code based upon E. F. Toro's weighted average flux (WAF) method for the investigation of air blast phenomenology. In support of that effort, we initially developed, tested and validated a one-dimensional shock code based upon Toro's weighted average flux (WAF) method. During development and testing of the one-dimensional shock code, we observed the following:

- Godunov's method accurately predicted the location of all discontinuities but the solution was smeared due to numerical viscosity
- application of WAF resulted in
 - decreased numerical viscosity
 - oscillations near discontinuities typical of second-order accurate methods
- application of TVD weight limiter functions adequately removed oscillations while maintaining behavior indicative of second-order methods
- among the weight limiter functions tested,
 - Super A resolved contact discontinuities best
 - van Leer A weight limiter function resulted in smooth regions behind shocks.

Before moving into two dimensions, we validated the one-dimensional shock code against experimental results obtained at Army Research Laboratory's 57cm shock tube facility. Here, we found the one-dimensional shock code underestimated the peak overpressure at the shock front by 6.8%.

Once the one-dimensional shock code was complete, we extended it into two spatial dimensions via Warming and Beam's variant of dimensional splitting. To verify proper implementation of this technique, we solved both Toro's cylindrical explosion problem and Sod's shock tube problem in two dimensions. Having shown the two-dimensional code worked properly, we modeled the detonation of a cylindrically symmetric explosive over an ideal surface to illustrate the capabilities of the code. We found the numerical solution modeled the physics of shock propagation and reflection well and that the transition from regular to Mach reflection occurred as expected. As a result of this effort, Air Force Institute of Technology now has a two-dimensional hydrodynamic shock code to further investigate air blast phenomenology.

Appendix A: Toro's Initial Guess Generator

Rapid convergence is achieved when $p_*^{(0)}$ is sufficiently close to the root. If $p_*^{(0)}$ is far from the actual root, the computational efficiency of ERS is severely degraded since much of the CPU time solving Equation (18), and therefore computational effort, is attributed to finding the root of equation (33). Worse, an extremely poor choice can result in a physically meaningless solution, $p_* \leq 0$. Clearly, success is dependent upon the numerical nature of Equation (30) and our choice of $p_*^{(0)}$.

Toro's pressure function, f , is monotonic and concave down because $f' > 0$ and $f'' < 0$. This means f is relatively benign. Therefore, the major concern here is not whether convergence is achieved but rather, how quickly it is achieved. Graphical inspection of the mathematical properties of Equation (30) provides some insight into how an appropriate $p_*^{(0)}$ might be chosen given the initial conditions and some assumptions.

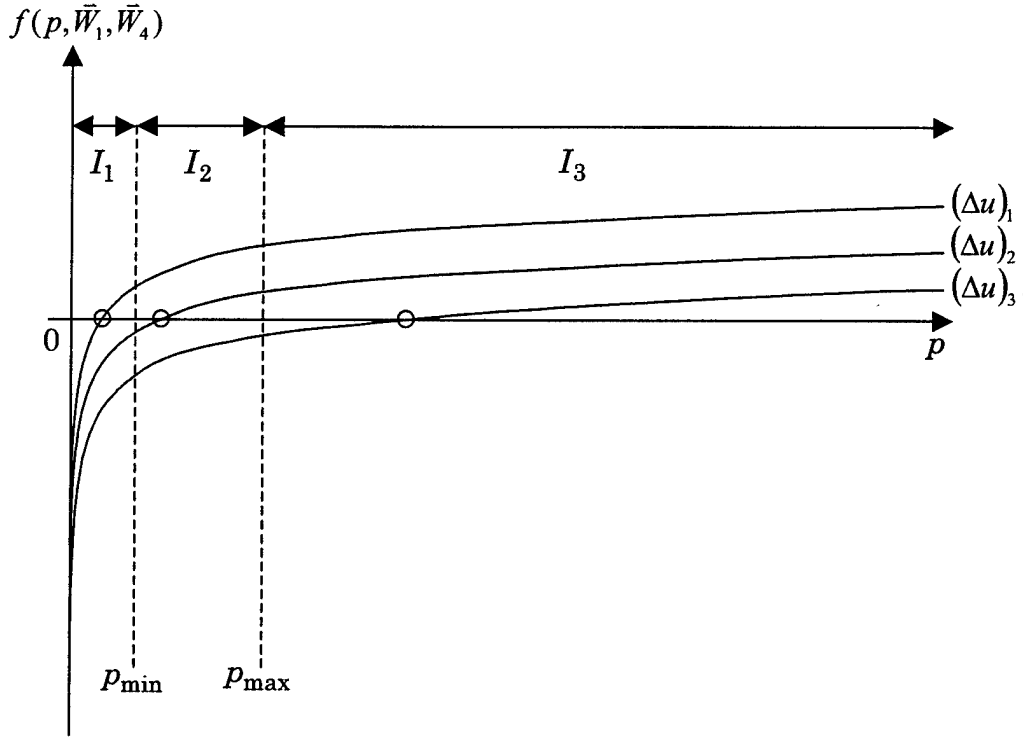


Figure 53: General Behavior of Toro's Pressure Function

Figure 53 illustrates the general behavior of the pressure function given initial conditions, \bar{W}_1 and \bar{W}_4 (Toro, 1997). Each curve in Figure 53 represents the pressure function given the same set of initial conditions for density and pressure. The difference between the curves is due to the particle velocity difference, where $\Delta u = u_4 - u_1$ varies.

With reference to Figure 53, the minimum and maximum pressures are defined to be:

$$p_{\min} = \min(p_1, p_4) \quad (132)$$

and

$$p_{\max} = \max(p_1, p_4). \quad (133)$$

These variables divide the pressure interval into three different subintervals:

$$I_1 : 0 < p \leq p_{\min},$$

$$I_2 : p_{\min} < p \leq p_{\max}$$

and

$$I_3 : p_{\max} < p \leq \infty.$$

Note that depending upon the value of Δu , the pressure function is translated up or down and the root changes accordingly.

For sufficiently large Δu the root lies within the first interval such that $p_* \leq p_{\min} < p_{\max}$. This indicates a two-rarefaction configuration. For $(\Delta u)_2$, p_* lies within the second interval where $p_{\min} < p_* \leq p_{\max}$. Then one of the outer waves is a rarefaction and the other a shock. Lastly, for small Δu , the root lies within the third interval where $p_* > p_{\max} > p_{\min}$. This indicates the two-shock configuration.

Given the general behavior of the pressure function and initial conditions, we can predict the general wave structure of the Riemann problem solution before it is solved. Armed with this knowledge, we can predict $p_*^{(0)}$ sufficiently close that the converged solution is typically obtained within ten iterations.

To determine where the root lies, we begin with the following assumptions:

- the root lies within the second interval.

- flow conditions are mild enough that a linearized approximation is sufficient.

With these assumptions, $p_*^{(0)}$ is obtained in the following manner (Toro, 1991):

$$p_*^{(0)} = \max(1.0 \times 10^{-6} P\alpha, p_{pvrs}) \quad (134)$$

where

$$p_{pvrs} = \frac{1}{2}(p_1 + p_4) + \frac{1}{8}(u_1 - u_4)(\rho_1 + \rho_4)(c_1 + c_4). \quad (135)$$

is taken from Toro's PVRs. Equation (134) is applied to ensure $p_*^{(0)} > 0$ because Equation (135) may, incorrectly, predict a negative pressure. Note that if strong shocks are present, our linearity assumption is no longer valid and PVRs does not perform well. Flow conditions are considered mild provided:

$$Q_S > \frac{p_{\max}}{p_{\min}} \quad (136)$$

where Q_S is a user-defined value typically set at 2 (Toro, 1991). Even if Equation (136) is false, we still calculate (134) for use in the two-shock approximation.

At this point, we check where $p_*^{(0)}$ lies in relation to p_{\min} and p_{\max} . If $p_{\min} < p_*^{(0)} \leq p_{\max}$ and flow conditions are sufficiently mild, $p_*^{(0)}$ is used as an initial guess in Equation (33). If $p_*^{(0)} \leq p_{\min}$, we assume the two-

rarefaction configuration and $p_*^{(0)}$ is obtained using the two-rarefaction approximation derived from Equation (30) (Toro, 1997):

$$p_*^{(0)} = \left(\frac{c_1 + c_4 - \frac{1}{2}(\gamma - 1)(u_4 - u_1)}{\frac{c_1}{p_1^{(\gamma-1)/2\gamma}} + \frac{c_4}{p_4^{(\gamma-1)/2\gamma}}} \right)^{1/2}. \quad (137)$$

Note that if two-rarefactions are actually present in the solution no iterations are required since this approximation is exact. If, however, $p_*^{(0)} > p_{\max}$ or

$Q_S < \frac{p_{\max}}{p_{\min}}$ is false, the two-shock configuration is assumed and the following

two-shock approximation is used (Toro, 1997):

$$p_*^{(0)} = \left[p_1 \left(\frac{2}{\rho_1(p_*^{(0)}(\gamma + 1) + p_1(\gamma - 1))} \right)^{1/2} + p_4 \left(\frac{2}{\rho_4(p_*^{(0)}(\gamma + 1) + p_4(\gamma - 1))} \right)^{1/2} + u_1 - u_4 \right] \times \left[\left(\frac{2}{\rho_1(p_*^{(0)}(\gamma + 1) + p_1(\gamma - 1))} \right)^{1/2} + \left(\frac{2}{\rho_4(p_*^{(0)}(\gamma + 1) + p_4(\gamma - 1))} \right)^{1/2} \right]^{-1} \quad (138)$$

Bibliography

- Abbott, Michael B. An Introduction To The Method of Characteristics. New York: American Elsevier, 1966.
- Advisory Group for Aeronautical Research and Development. Fundamental Data Obtained From Shock-Tube Experiments. Ed. Antonio Ferri. New York: Pergamon Press, 1961.
- Anderson, John D. Modern Compressible Flow with Historical Perspective. New York: McGraw-Hill, 1982.
- Baker, Wilfred E. Explosions in Air. Austin TX: University of Texas Press, 1973.
- Ben-Dor, Gabi. Shock Wave Reflection Phenomena. New York: Springer-Verlag, 1992.
- Bradley, John N. Shock Waves in Chemistry and Physics. London: Methuen & Co Ltd., 1962.
- Burden, Richard L. and J. Douglas Faires. Numerical Analysis: Fifth Edition. Boston: PWS-Kent Publishing Company, 1993.
- Chang, Tung and Ling Hsiao. The Riemann Problem and Interaction of Waves in Gas Dynamics. New York: John Wiley and Sons, Inc., 1989.
- Courant, R. and K. O. Friedrichs. Supersonic Flow and Shock Waves. New York: Interscience Publishers, Inc., 1948.
- Einfeldt, B. and others. "On Godunov-Type Methods for Gas Dynamics," SIAM Journal of Numerical Analysis, 92: 273-295 (1991).
- Giles, Michael B. "Nonreflecting Boundary Conditions for Euler Equation Calculations," American Institute of Aeronautics and Astronautics, 28: 2050-2058 (1990).
- Glasstone, Samuel and Philip J. Dolan. The Effects of Nuclear Weapons. (Third Edition). Published by the US Department of Defense and the Energy Research and Development Administration, US Government Printing Office, 1977 (reprinted by the Defense Special Weapons Agency now the Defense Threat Reduction Agency).

- Hirsch, Charles. Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows. Chichester, England: John Wiley and Sons, 1990.
- Kinney, Gilbert F and Kenneth J Graham. Explosive Shocks in Air. (Second Edition). Berlin: Springer-Verlag, 1985.
- Lax, Peter D. Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves. (Third Edition). Montpelier, VT: Capital City Press, 1990.
- LeVeque, Randall J. Numerical Methods for Conservation Laws. (Second Edition). Basel: Birkhauser Verlag, 1992.
- Rizzi, A. "Numerical Implementation of Solid-Body Boundary Conditions for the Euler Equations," Zeitschrift für Angewandte Mathematik und Mechanik, 58: 301-304 (1978).
- Roe, Philip L. "Some Contributions to the Modelling of Discontinuous Flows," in Large-Scale Computations in Fluid Mechanics. Ed. Bjorn E. Engquist and others. Providence, Rhode Island: American Mathematical Society, 1985.
- Schraml, Stephen J., A One-Dimensional Numerical Method for Simulating Multiple Stage Shock Tube Flows. Army Research Laboratory, Aberdeen Proving Ground, MD November 1996 (ARL-TR-1255).
- Shi, J. and E. F. Toro. "Fully Discrete High-Order Shock-Capturing Numerical Schemes," International Journal for Numerical Methods in Fluids, 23: 241-269 (1996).
- Sod, Gary D. "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," Journal of Computational Physics, 27: 1-31 (1978).
- Sweby, P. K. "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," SIAM Journal of Numerical Analysis, 21: 995-1011 (1984).
- Thompson, Kevin W. "Time Dependent Boundary Conditions for Hyperbolic Systems," Journal of Computational Physics, 68: 1-24 (1987).

- Toro, E. F. "A Fast Riemann Solver with Constant Covolume applied to the Random Choice Method," International Journal for Numerical Methods in Fluids, 9: 1145-1164 (1989).
- , "A Linearized Riemann Solver for the Time-Dependent Euler Equations of Gas Dynamics," Proceedings of the Royal Society of London, A424: 683-693 (1991).
- , Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Berlin: Springer-Verlag, 1997.
- , "The Weighted Average Flux Method Applied to the Euler Equations," Philosophical Transactions of the Royal Society of London, A341: 499-530 (1992).
- , "A Weighted Average Flux Method for Hyperbolic Conservation Laws," Proceedings of the Royal Society of London, A423: 401-418 (1989).
- van Albada, G. D. and others. "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," Astron. Astrophysics, 108: 76-84 (1982).
- van Leer, Bram. "Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second Order Scheme," Journal of Computational Physics, 14: 361-370 (1974).
- Warming, R. F. and R. W. Beam "Upwind Second Order Difference Schemes with Applications in Aerodynamic Flows," American Institute of Aeronautics and Astronautics, 24: 1241-1249 (1976).
- Woodward, P. and P. Colella "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," Journal of Computational Physics, 54: 115-173 (1984).

VITA

Mark P. Wittig was born on 2 April 1966 in Wichita, Kansas. He graduated from Winnetonka High School in Kansas City, Missouri, in 1984. He attended the University of Missouri at Kansas City, graduating in May of 1989 with a degree of Bachelor of Science in Biology. In September of that same year, he attended the air defense artillery officer basic course and HAWK weapon track at Ft Bliss, Texas. Upon completion of ADA OBC, he was assigned to the 4th Battalion, 1st Air Defense Artillery Regiment in the Federal Republic of Germany where he served as a HAWK assault fire platoon leader, tactical control officer and tactical director. In 1993, he returned to FT Bliss, Texas where he successfully completed both the officer advanced course and forward area air defense officer course. In 1994, he relocated to FT Hood, Texas. After an initial stint as the operations and plans officer for the 31st Air Defense Artillery Brigade, he took command of Delta Battery, 2nd Battalion, 5th Air Defense Artillery Regiment, 2nd Armored Division which was later reflagged as Delta Battery, 1st Battalion, 44th Air Defense Artillery Regiment, 4th Infantry Division. He remained in command until entering the School of Engineering, Air Force Institute of Technology in 1997. Upon graduation, he will be reassigned to the Defense Threat Reduction Agency in Washington, D.C., where he will serve in the stockpile technical support office.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Jun 1999		3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE Implementation of a Two-Dimensional Hydrodynamic Shock Code Based Upon the Weighted Average Flux Method			5. FUNDING NUMBERS	
6. AUTHOR(S) CPT Mark P. Wittig				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFIT/ENP 2950 P Street Wright-Patterson AFB, OH 45433-7765 Advisor: Dr. Kirk A. Mathews			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAP/ENP/99J-02	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Threat Reduction Agency ATTN: LTC Robert Mull 6801 Telegraph Rd Alexandria, VA 22310			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Distibution Unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (Maximum 200 words) Numerical modeling of shock propagation and reflection is of interest to the Department of Defense (DoD). Propriety state-of-the-art codes based upon E. F. Toro's weighted average flux (WAF) method are being used to investigate complex shock reflection phenomena. Here we develop, test, and validate a one-dimensional hydrodynamic shock code. We apply WAF to Gudonov's first-order upwind method to achieve second-order accuracy. Oscillations, typical of second-order methods, are then removed using adaptive weight limiter functions based upon total variation diminishing (TVD) flux limiters. An adaptive Riemann solver routine is also implemented to improve computational efficiency. This one-dimensional code is then extended into two dimensions via Warming and Beam's variation on dimensional splitting. The numerical capabilities of the two-dimensional code are demonstrated by modeling the detonation of a cylindrically symmetric explosive with the axis of the cylinder oriented horizontally above an ideal surface.				
14. SUBJECT TERMS Shock Waves, Blast Waves, Computational Fluid Dynamics			15. NUMBER OF PAGES 147	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unc	